

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Сергій СТИПЕНКО

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Комп'ютерні системи та мережі»**

**спеціальності 123 «Комп'ютерна інженерія»**

**на тему: «Система цифрового радіозв'язку»**

Виконав:

студент IV курсу, групи ІО-62

Коробейніков Дмитро Юрійович

\_\_\_\_\_

Керівник:

доктор технічних наук, доцент,

Сергієнко Анатолій Михайлович

\_\_\_\_\_

Консультант з нормаконтролю:

доктор технічних наук, професор,

Сімоненко Валерій Павлович

\_\_\_\_\_

Рецензент:

\_\_\_\_\_

Засвідчую, що у цій дипломній роботі  
немає запозичень з праць інших авторів  
без відповідних посилань.

Студент (-ка) \_\_\_\_\_

Київ – 2020 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра обчислювальної техніки**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Сергій СТИПЕНКО

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Коробейнікову Дмитру Юрійович**

1. Тема проєкту «Цифрова система радіозв'язку», керівник проєкту Сергієнко Анатолій Михайлович, доктор технічних наук, доцент, затверджені наказом по університету від «\_\_\_» \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін подання студентом проєкту \_\_\_\_\_

3. Вихідні дані до проєкту

Технічна документація, теоретичні дані, інтернет-публікації за темою роботи

4. Зміст пояснювальної записки

Провести аналіз особливостей систем цифрового радіозв'язку;

Провести огляд стандартів реалізації систем цифрового радіозв'язку;

Розробити програмну реалізацію одного з стандартів на базі плати Lime SDR;

Провести симуляцію роботи створеної системи та проаналізувати її результати;

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

Креслення: «Загальна схема системи. Схема даних. Схема структурна», «Алгоритм CORDIC. Схема алгоритму», «Формувач скрамблюючого коду. Схема функціональна», презентація.

6. Дата видачі завдання \_\_\_\_\_

#### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Затвердження теми роботи	10.12.2019	виконано
2	Вивчення та аналіз завдання	13.12.2019-02.02.2020	виконано
3	Аналіз особливостей систем цифрового радіозв'язку та засобів їх реалізації	03.02.2020-01.03.2020	виконано
4	Огляд стандартів цифрового радіозв'язку	02.03.2020-15.04.2020	виконано
5	Розробка програмної реалізації стандарту цифрового радіозв'язку	16.04.2020-13.05.2020	виконано
6	Проведення симуляції роботи створенної системи та аналіз результатів	14.05.2020-19.05.2020	виконано
7	Оформлення матеріалів роботи	20.05.2020-24.05.2020	виконано
8	Передзахист	05.06.2020	виконано
9	Захист	15.06.2020	виконано

Студент

Дмитро КОРОБЕЙКОВ

Керівник

Анатолій СЕРГІЄНКО

## **Анотація**

Дана робота присвячена реалізації системи цифрового радіозв'язку на базі програмованої плати Lime SDR. Актуальність даної задачі зумовлена повсюдним розповсюдженням радіозв'язку, через що повстають проблеми стороннього втручання в передачу, необхідності передачі конфіденційної інформації, необхідності розміщення багатьох абонентів на одній і тій самій частоті, тощо. Данні проблеми вирішуються за допомогою впровадження цифрових маніпуляцій з інформацією, що передається.

Реалізована система відтворює протокол WCDMA, котрий являється реально використовуваним протоколом зв'язку, що дає змогу розвинути ідею даної роботи до певного фізичного прототипу чи комерційного продукту. Крім того, розроблена система являється програмно-визначеною, що дозволяє при необхідності швидко змінювати використовуваний стандарт, а також проводити тонке налаштування системи без внесення змін в апаратну частину, що не можливо в апаратно-визначених системах.

## **Abstract**

This work is dedicated to the implementation of a digital radio communication system based on the programmable Lime SDR board. The urgency of this task is due to the widespread use of radio communications, which raises the problem of third-party interference in the transmission, the need to transmit confidential information, the need to place many subscribers on the same frequency, and so on. These problems are solved through the introduction of digital manipulation of transmitted information.

The implemented system reproduces the WCDMA protocol, which is a actually used communication protocol, which allows to develop the idea of this work to a specific physical prototype or commercial product. In addition, the developed system is software-defined, which allows, if necessary, to quickly change the standard used, as well as to fine-tune the system without making changes to the hardware, which is not possible in hardware-defined systems.



## Технічне завдання до дипломного проекту

### ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ .....	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ .....	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	2
5.1. Вимоги до розроблюваного продукту.....	2
5.2. Вимоги до програмного забезпечення .....	3
5.3. Вимоги до апаратного забезпечення .....	3

					ІА/Ц.467100.002 ТЗ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Карабейніков Д. Ю.			Система цифрового радіозв'язку  Технічне завдання	Літ.	Аркуш
Перевір.		Сергієнко А. С.				1	3
						НТУУ "КПІ", ФІОТ, ІО-62	
Н. контр.		Сімоненко В. П.					
Затверд.							

## 1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання розповсюджується на розробку системи цифрового радіозв'язку на базі ПЛІС Lime SDR.

Область застосування: розробка програмно-визначених пристроїв цифрового радіозв'язку на програмованих логічних інтегральних схемах.

## 2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки служить завдання на виконання розробки системи цифрового радіозв'язку, затвердженого кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний Інститут».

## 3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проекту є розробка системи цифрового радіозв'язку на базі програмованої плати Lime SDR.

## 4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами для розробки служать науково-технічна література з комп'ютерних технологій, публікації в періодичних виданнях, довідники з програмованих логічних інтегральних схем, публікації в Інтернеті за даним питанням.

## 5. ТЕХНІЧНІ ВИМОГИ

### 5.1. Вимоги до розроблюваного продукту

- Реалізація одного з існуючих протоколів зв'язку.
- Реалізувати перетворення сигналу таким чином, щоб процес можливо було обернути на приймачі.
- Виконання симуляції роботи системи засобами середовища Active HDL.

					ІА/ЛЦ.467100.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

## 5.2. Вимоги до програмного забезпечення

- Операційна система Windows 7, MS Windows 8/8.1, MS Windows 10
- Active HDL

## 5.3. Вимоги до апаратного забезпечення

- Комп'ютер на базі 32-розрядного процесору 1ГГц і швидше
- Оперативної пам'яті не менше 1 Гбайт

					ІАЛЦ.467100.002 ТЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		



**Пояснювальна записка  
до дипломного проєкту  
на тему: «Система цифрового радіозв'язку»**

Київ – 2020 року

## ЗМІСТ

ВСТУП .....	4
РОЗДІЛ 1 .....	6
ОГЛЯД СИСТЕМ ЦИФРОВОГО РАДІОЗВ'ЯЗКУ .....	6
1.1. Принципи роботи цифрового радіозв'язку .....	6
1.1.1 Цифрова модуляція .....	7
1.1.2 Автокореляція .....	9
1.1.3 Шифрування каналу .....	11
1.2. Особливості SDR .....	11
ВИСНОВКИ ДО РОЗДІЛУ 1 .....	14
РОЗДІЛ 2 .....	15
АНАЛІЗ СТАНДАРТІВ ЦИФРОВОГО РАДІОЗВ'ЯЗКУ .....	15
2.1 Стандарти односторонньої передачі .....	15
2.1.1 Стандарт Digital Audio Broadcasting (DAB) .....	15
2.1.1 Стандарт DAB+ .....	17
2.1.3 Стандарт Integrated Services Digital Broadcasting (ISDB) .....	19
2.2 Стандарти двостороннього зв'язку .....	20
2.2.1 Стандарт CDMA2000 .....	21
2.2.2 Стандарт UMTS-TDD .....	24
2.2.3 Стандарт WCDMA .....	26
ВИСНОВКИ ДО РОЗДІЛУ 2 .....	33
РОЗДІЛ 3 .....	34
ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ СИСТЕМИ .....	34
3.1 Опис програмованої плати та її основних компонентів .....	34
3.2 Опис алгоритму та компонентів програми .....	41
3.2.1 Модуль main_controller .....	41
3.2.2 Модуль mixer .....	41

3.2.3 Модуль modulator .....	42
3.2.4 Модуль scrambler .....	43
3.2.5 Допоміжні модулі cordic та sin_cos .....	44
ВИСНОВКИ ДО РОЗДІЛУ 3 .....	47
РОЗДІЛ 4 .....	48
СИМУЛЯЦІЯ РОБОТИ СИСТЕМИ .....	48
4.1 Результати симуляції виконання програми та аналіз її результатів .....	48
4.2 Інструкція користувача .....	49
ВИСНОВКИ ДО РОЗДІЛУ 4 .....	52
ВИСНОВКИ .....	53
ПЕРЕЛІК ПОСИЛАНЬ .....	54

## ВСТУП

Через повсемістне розповсюдження аналогового радіозв'язку виникла проблема досягнення технічних лімітів технології, що виражається в перевантаженні каналів, недостатній ємності передачі, або загрозі стороннього втручання під час передачі. В зв'язку з цим виникла технологія цифрового радіозв'язку, що покликана виправити наведені вище недоліки.

### Актуальність теми

З огляду на достатньо широке розповсюдження згаданих вище систем постає проблема наявності великої кількості можливих для використання стандартів зв'язку, котрі, однак, не являються універсальними, і потребують підбору под конкретну задачу та апаратну реалізацію. До того ж реалізація цифрового радіо саме у вигляді програмно-визначеної системи відкриває масу вигод, недоступних апаратно-визначеним аналогам.

### Мета і задачі дослідження

Метою данної роботи є розробка системи цифрового радіозв'язку на базі плати Lime SDR, що реалізує один з стандартів цифрового радіозв'язку.

Для досягнення поставленої мети були поставлені наступні основні задачі:

- Провести аналіз можливих технологій цифрового радіозв'язку, виявити їх переваги та недоліки,;
- Розробити проект та реалізацію системи, що використовує певний стандарт цифрового зв'язку;
- Провести моделювання системи за допомогою VHDL-симулятора;
- Проаналізувати результати моделювання системи.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

### Практичне значення

Реалізована система дозволяє передавати інформацію шляхом радіозв'язку за стандартом WCDMA, при цьому являється програмно-визначеною системою.

					ІА/Ц.467100.003 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 1

### ОГЛЯД СИСТЕМ ЦИФРОВОГО РАДІОЗВ'ЯЗКУ

Радіозв'язок являє собою передачу інформації через простір, шляхом трансформування електричних сигналів в радіохвилі певної довжини. В свою чергу, системи, що реалізують радіозв'язок, підрозділяються на два великих класи – аналогові та цифрові. В аналогових системах інформація передається без внесення в неї будь-яких коректив, прямим перетворенням електричного сигналу в радіохвилі шляхом подачі електричного струму на модулятор [1]. В цифрових системах сигнал піддається додатковій обробці, і передається у вигляді цифрової інформації [2].

Крім того, існує подальше ділення систем цифрового радіозв'язку. Класичним рішенням є апаратно-визначені системи, в яких сам процес формування сигналу на передачу відбувається за рахунок вбудованих в систему компонентів. Такі системи не є гнучкими, і як можливе рішення проблеми виникли програмно-визначені системи, або ж SDR (software-defined radio). В таких системах функції, зазвичай виконувемі структурними компонентами, виконує програмне забезпечення. Це дозволяє достатньо просто змінювати стандарт передачі, а також дозволяє швидко налаштовувати систему, не змінюючи її фізичні компоненти [3].

#### 1.1. Принципи роботи цифрового радіозв'язку

Загалом, цифровий радіозв'язок використовується для досягнення одного з наступних ефектів:

- *захисту каналу зв'язку від стороннього втручання*
- *передачі кількох каналів на одній й тій самій частоті*
- *досягнення кращої якості передачі інформації*

Для виконання даних умов в цифровому радіозв'язку застосовуються наступні методи:

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

### 1.1.1 Цифрова модуляція

В аналоговому радіозв'язку модуляція означає «продавлювання» моделюємого сигналу модулюючим, що дає змогу отримати сигнал необхідної амплітуди для передачі[1].

В цифрових системах модельція виконується за допомогою математичних методів, шляхом маніпуляцій з двійковим представленням інформації. Далі наведено короткий опис чотирьох типів цифрової модуліції.

Фазовий зсув передає дані шляхом зміни фази опорного сигналу постійної частоти (рис. 1.1). Модуляція здійснюється шляхом зміни синусоїдальних та косинусних входів у точний час[2].

Даний тип модуляції відносно простий для реалізації, особливо в порівнянні з квадратурною модуляцією, опис якої наведено нижче. Через це даний спосіб достатньо широко використовується в сучасних системах.

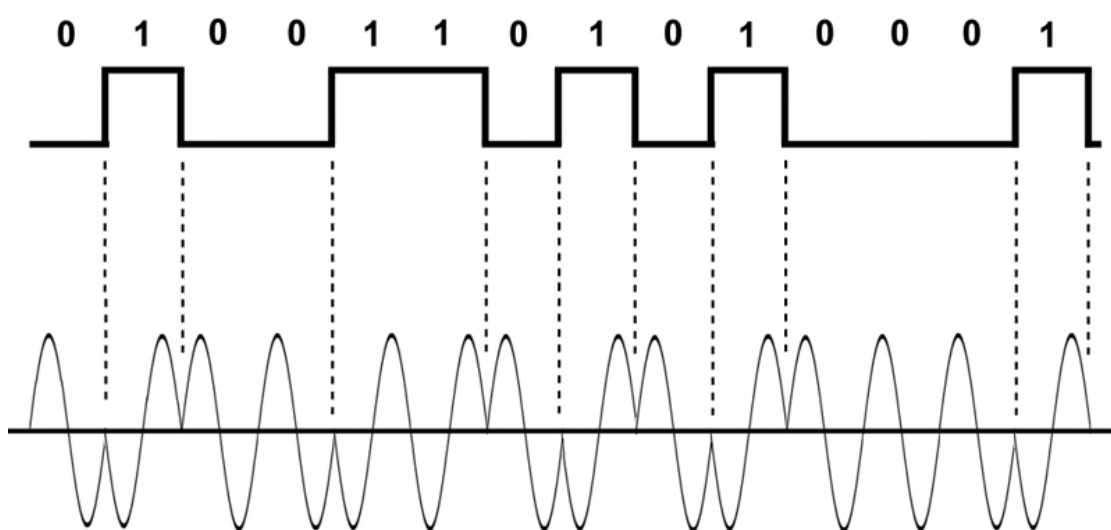


Рис. 1.1: схематичне зображення фазової модуляції

При частотній модуляції стрибкоподібно змінюється частота несучого сигналу в залежності від значень символів інформаційної послідовності[2]. Схематичне зображення роботи даного типу модуляції приведено на рис. 1.2. Частотна модуляція достатньо стійка до перешкод, оскільки перешкоди, в основному, впливають на амплітуду сигналу, а не на його частоту.

Частотної модуляцією з безперервною фазою називається спосіб частотної модуляції, при якому фаза послідовності радіосигналів є безперервною, тобто не має стрибків.

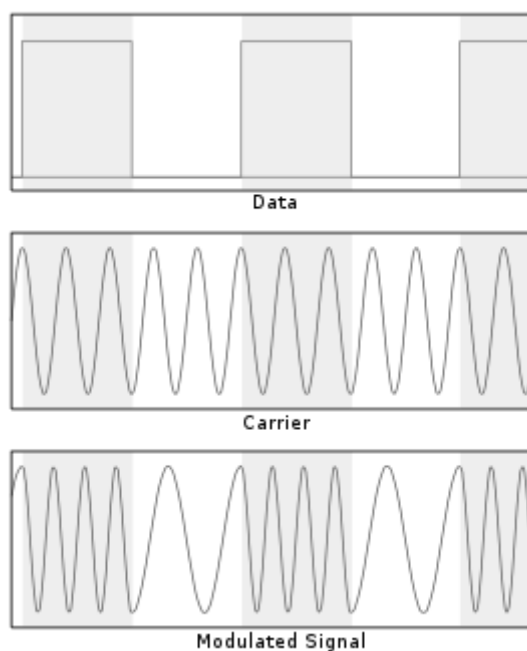


Рис. 1.2: схематичне зображення частотної модуляції

Амплітудна модуляція, в свою чергу, базується на скачкоподібній зміні амплітуди несучого сигналу, в залежності від символу інформаційної послідовності[2].

У данній системі бінарний символ 1 представлений передачею хвилі несучої з фіксованою амплітудою та фіксованою частотою протягом бітної тривалості  $T$  секунд. Якщо значення сигналу дорівнює 1, то сигнал несучої передаватиметься, в іншому випадку буде передано значення сигналу 0 (рис. 1.3).

Будь-яка схема цифрової модуляції використовує кінцеву кількість різних сигналів для представлення цифрових даних. Амплітудна модуляція використовує кінцеву кількість амплітуд, кожній присвоєній унікальний зразок двійкових цифр. Зазвичай кожна амплітуда кодує рівну кількість бітів. Кожна послідовність бітів утворює символ, який представлений певною амплітудою. Демодулятор, який призначений спеціально для символу-набору, використовуваного модулятором, визначає амплітуду сигналу і



відображає його назад до символу, відновлюючи, таким чином, початкові данні.

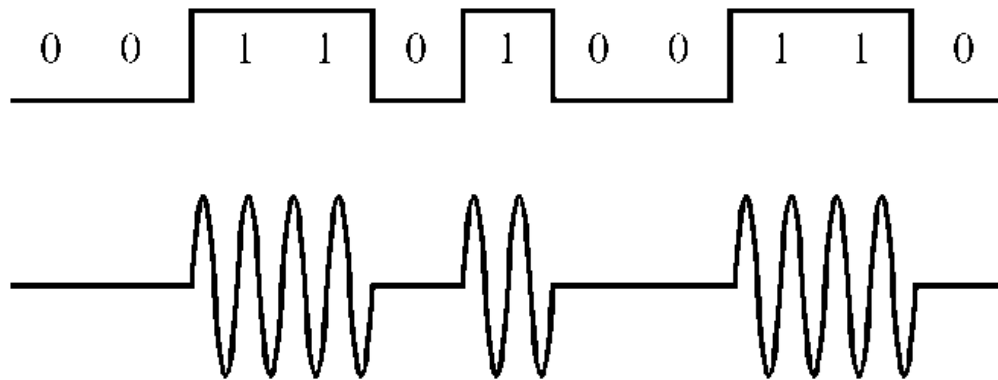


Рис. 1.3: схематичне зображення амплітудної модуляції

Квадратурна модуляція передає два аналогових сигналу повідомлення або два цифрових бітових потоку, змінюючи (модулюючи) амплітуди двох несучих хвиль, використовуючи схему цифрової модуляції амплітудного зсуву або аналогову схему модуляції амплітуди[2]. Дві несучі хвилі однакової частоти поза фазою, один з одним на  $90^\circ$ , умова, відома як ортогональність або квадратура. Переданий сигнал створюється додаванням двох несучих хвиль разом (рис. 1.4). На приймачі обидві хвилі можуть бути когерентно розділені (демодульовані) через властивість їх ортогональності. Ще однією ключовою властивістю є те, що модуляції - це хвилі низької частоти порівняно з частотою несучої, яка відома як вузькосмугова припущення.

### 1.1.2 Автокореляція

Автокорреляція - статистичний взаємозв'язок між випадковими величинами з одного ряду, але узятих із зсувом[2].

Для реалізації данного принципу застосовується наступна функція:

$$\Psi(\tau) = \int f(t)f(t - \tau)dt \quad (1.1)$$

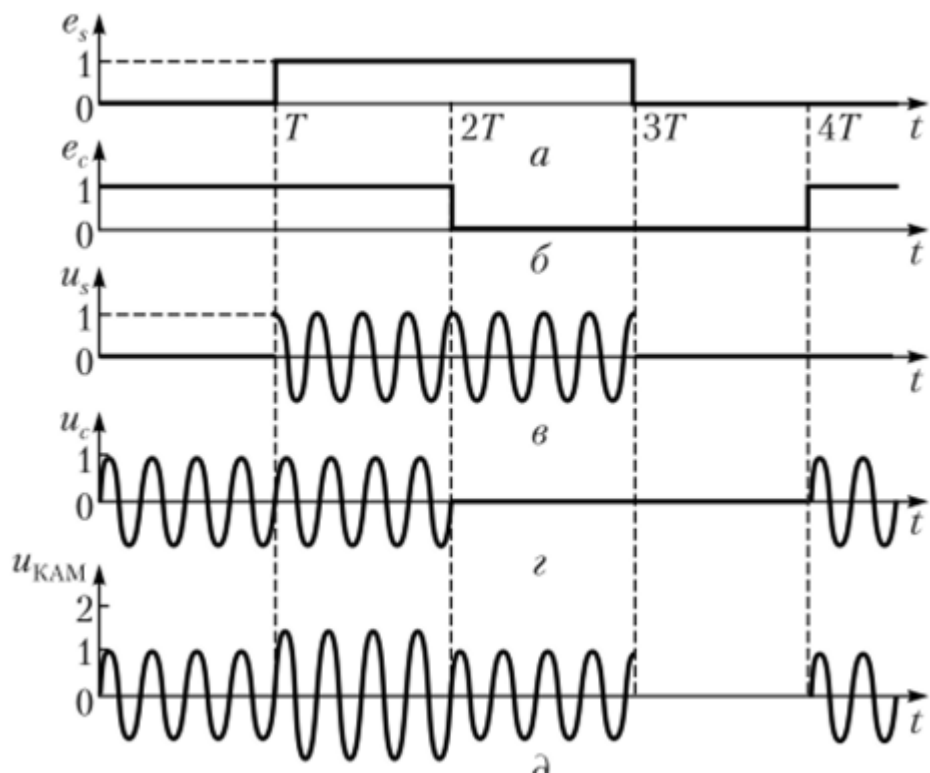


Рис. 1.4: схематичне зображення квадратурної моделювання

Тобто відбувається накладання функції самої на себе, але зі ссувом. Данна маніпуляція необхідна для уникнення виникнення помилок при передачі даних.

Тобто, у той час, коли вхідний сигнал найбільше схожий на потрібну нам функцію - функція кореляції матиме пік. Ширина цього піку, без урахування шуму, буде дорівнює вдвічі довжині імпульсу зонда і буде симетричною щодо центрального піку - навіть якщо досліджуваний сигнал не симетричний.

Метод кореляції є оптимальним методом визначення сигналу відомої форми на тлі білого шуму - тобто метод має найкраще співвідношення сигнал/шум[4]. Імпульс зонда повинен відповідати наступним вимогам - мати максимально вузький центральний пік і в той же час мати мінімальний рівень бічних пелюсток, тобто функція схожа на себе лише за дуже короткий проміжок часу - для зміни мало і це не схоже.

### 1.1.3 Шифрування каналу

Власне, являє собою шифрування даних, що передаються каналом. При використанні в системах потокового зв'язку зазвичай застосовуються поточкові методи шифрування, для уникнення затримок при передачі сигналу.

Використання шифрування сигналу може бути зумовлено різними факторами. окрім очевидного захисту даних від стороннього втручання певні типи шифрування дають змогу розміщувати декілька абонентів на одній частоті, при цьому уникаючи перекривання сигналів між собою за рахунок того, що кожен з абонентів розшифровує лише призначений для нього набір даних[4].

Проте такий підхід пов'язаний з певними недоліками, адже алгоритми потокового шифрування не відрізняються великою криптостійкістю, а також зазвичай потребують спеціальний процесор для своєї роботи[5].

Крім того, при використанні кодування за допомогою кодів Ріда-Соломона з'являється можливість корегувати втрачені частини повідомлення, що по якійсь причині не дійшли до приймача. Це зумовлено самою структурою даних кодів[4].

### 1.2. Особливості SDR

В SDR компоненти, традиційно реалізовувані в якості апаратних засобів (наприклад , змішувачі , фільтри , підсилювачі , модулятори / демодулятор , детектори і т.д.), реалізовані за допомогою програмного забезпечення на персональному комп'ютері або вбудованій обчислювальній системі[3].

Базова система SDR може складатися з персонального комп'ютера, або іншого аналого-цифрового перетворювача, до якого підключена деяка система випромінювання радіохвиль. Значна кількість обробки сигналів передається процесору загального призначення, а не в апараті спеціального призначення (електронних схемах). Така конструкція створює радіо, яке може приймати та передавати дуже різні радіопротоколи, обумовлені виключно на використаному програмному забезпеченні[5].

					ІА/ЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

Програмні радіостанції широко використовуються у службах військового зв'язку та стільникового зв'язку, так як в обох галузях вони повинні обслуговувати широкий спектр змінних протоколів радіо в реальному часі[5].

Загальна схема SDR наведена на рис. 1.5

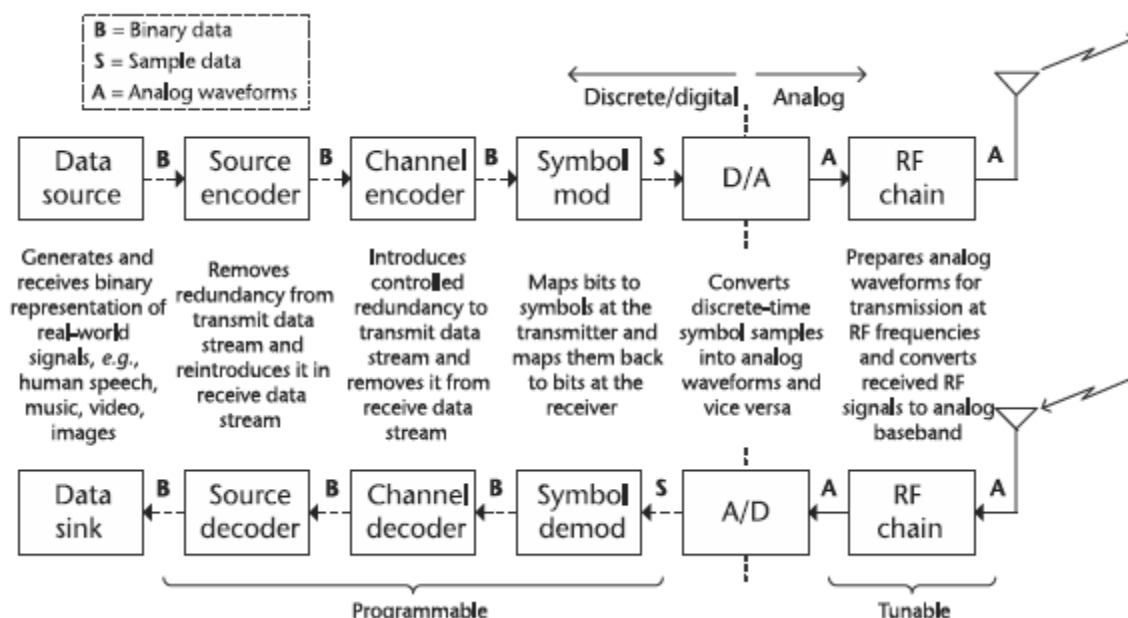


Рис. 1.5: загальна схема SDR

Радіо, визначене програмним забезпеченням, може утворювати радіовузол у будь-якому, від посилання до точки до елемента у великій спеціальній мережі радіостанцій. Передній кінець радіоприймача, є єдиним компонентом у більшій системі зв'язку, яка, як правило, включає зовнішні фільтри та специфічні для смуги антени[5]. Основними аспектами цього є

- Аналоговий радіочастотний розділ (антенна, радіочастотні фільтри, вхідний мюкс, LNA, посилення, загасання, змішувач);
- Аналоговий розділ базової смуги (аналогові фільтри, АЦП або ЦАП);
- Деякі блоки обробки сигналів (фіксовані фільтри всередині приймача, або визначені користувачем всередині FPGA або DSP, або процесор загального призначення).

З огляду на наведені вище данні було обрано матеріальну базу для реалізації предмету роботи, що являє собою проектну плату Lime SDR виробництва Lime Microsystems. В данній платі наявний програмований

логічний керуючий блок, та два приймача-передавача[6]. Така конфігурація дозволяє запрограмувати апаратний комплекс необхідним чином, і реалізувати SDR на його базі.

					ІАЛЦ.467100.003 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ ДО РОЗДІЛУ 1

1. Особливості систем цифрового радіозв'язку, а також системні особливості програмно-визначених систем радіозв'язку диктують вибір апаратної реалізації, придатної для втілення предмету данної роботи
2. ПЛІС Lime SDR дозволяє реалізувати програмно-визначену систему цифрового радіозв'язку по причині наявності на платі компонентів, необхідних для створення подібної системи, а саме програмованого логічного обчислювального пристрою, а також радіопередавача.

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

## РОЗДІЛ 2

### АНАЛІЗ СТАНДАРТІВ ЦИФРОВОГО РАДІОЗВ'ЯЗКУ

З метою впорядкування технологій цифрового радіозв'язку було введено різноманітні стандарти, що відрізняються за способом реалізації та сферою застосування. В данному розділі наведено короткий огляд різноманітних стандартів односторонньої та двусторонньої передачі, з метою визначення найбільш підходящих для розробки.

#### 2.1 Стандарти односторонньої передачі

Данні стандарти призначені для передачі даних від передовача до приймача, але не у зворотньому напрямку. Використання таких стандартів може бути зумовлено технічними обмеженнями обладнання на стороні приймача, що не дозволяє реалізувати зворотній зв'язок, або специфікою сфери застосування[4]. На практиці застосовуються у системах радіомовлення та телемовлення, але в данній роботі, з огляду на обмеження апаратної частини, будуть розглянуті лише перші.

##### 2.1.1 Стандарт Digital Audio Broadcasting (DAB)

Стандарт DAB був створений як європейський дослідницький проект у 1980-х роках[4]. Даний стандарт використовує широкосмугову технологію широкомовної передачі, і для неї зазвичай виділяються спектри в діапазоні 174-240 МГц та 1.452–1.492 ГГц, хоча система передбачає можливість роботи від 30 до 300 МГц.

З точки зору стакування протоколу моделі OSI, технології, що застосовуються в DAB, містять такі шари: аудіокодек MPEG-1 Audio Layer II займає презентаційний шар[8]. Нижче знаходиться рівень зв'язку даних, який відповідає за мультиплексування статистичного поділу часу та синхронізацію кадрів. Нарешті, фізичний рівень містить кодування виправлення помилок, модуляцію OFDM та стосується передачі та прийому даних по повітрю[4].

DAB використовує проколоте згорткове кодування для свого ECC[8]. Схема кодування використовує неоднаковий захист від помилок (UEP), що

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

означає, що частини аудіо-бітового потоку, більш чутливі до помилок, що спричиняють чутні порушення, забезпечуються більш високим захистом (тобто меншою швидкістю кодування) і навпаки. Однак схема UEP, що використовується в DAB, призводить до того, що між користувачем, який відчуває гарну якість прийому, і користувачем, який взагалі немає прийому, є сіра зона, на відміну від ситуації з більшістю інших бездротових цифрових систем зв'язку, які мають різкий "цифровий обрив", де сигнал швидко стає непридатним, якщо сила сигналу падає нижче певного порогу[4]. Коли слухачі DAB отримують сигнал у цій проміжній області сили, вони відчувають «булькаючий» звук, який перериває відтворення звуку.

Імунітет до завмирання та міжсимвольних перешкод (викликаних багатоплановим розповсюдженням) досягається без вирівнювання за допомогою методів модуляції OFDM та DQPSK[8].

Використовуючи значення для режиму I передачі (TM I), модуляція OFDM складається з 1536 піднесучих, які передаються паралельно[8]. Корисна частина періоду символів OFDM - 1 мілісекунда, в результаті чого несучі OFDM мають кожен пропускну здатність 1 кГц через зворотну залежність між цими двома параметрами, а загальна пропускну здатність каналу OFDM становить 1537 кГц[8]. Інтервал захисту OFDM для TM I становить 246 мікросекунд, що означає, що загальна тривалість символу OFDM становить 1,224 мілісекунди. Тривалість захисного інтервалу також визначає максимальний поділ між передавачами, які є частиною тієї ж одночастотної мережі (SFN), що становить приблизно 74 км для TM I[8].

DAB - це ефективна система мовлення (наприклад, система "один на багато"), здатна надавати надійні цифрові послуги всім користувачам, які знаходяться в зоні покриття в режимі реального часу. Він особливо підходить для прийому мобільними та переносними приймачами та в місцях, де пряма лінія огляду між передавачем та приймачем неможлива[4].

З іншого боку, мобільний телефонний зв'язок GSM / 3G UMTS / 4G і далі більше підходить для надання медіа-послуг на замовлення

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16



індивідуальним клієнтам або відносно невеликим групам клієнтів[8]. Телекомунікаційні системи технічно здатні надавати послуги декільком користувачам одночасно, за умови, що кількість одночасних користувачів (або, іншими словами, загальна пропускна здатність) не перевищує потужність, через що мережа швидко руйнується. Крім того, використання телекомунікаційних послуг за сценарієм "один на багато" є значно дорожчим для мовника та слухача, ніж використання безкоштовних ефірних систем DAB в ефірі[8].

Глобальний галузевий форум WorldDMB наразі доповнює ряд стандартів, що впроваджують стандартизовані способи забезпечення розширеної мультимедійної та Інтернет-функціональності та інтерактивності.

### **2.1.1 Стандарт DAB+**

WorldDMB , організація, відповідальна за стандарти DAB, оголосила DAB +, що було значно оновлено до стандарту DAB в 2006 році, коли було прийнято аудіокодек HE-AAC v2 (також відомий як eAAC + ) [8]. AAC + використовує алгоритм модифікованого дискретного косинусного перетворення (MDCT)[4]. Новий стандарт, який називається DAB +, також прийняв формат звуку MPEG Surround та більш чітке кодування виправлення помилок у вигляді кодування Ріда-Соломона.

DAB + DMB - це широкосмугова технологія, що вимагає мультиплексування послуг перед передачею. Використання смуг УКХ та УВЧ означає, що послуги DAB + зазвичай передаються з високих майданчиків, таких як вершини пагорбів, будинки чи вежі[8].

Загалом, послуги DAB +, ймовірно, будуть розташовуватися разом із існуючими телевізійними передачами та послугами FM-телебачення, враховуючи витрати на розробку нових місць передачі та зростаючі труднощі з отриманням схвалення місцевою радою щодо планування нових ділянок передачі. Особливо важливо спільно знайти DAB + з послугами високої потужності телебачення VHF діапазону III, щоб мінімізувати проблеми перешкод[4].

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

Станції, які спочатку транслювали більше однієї програми AM / FM з того самого передавача, можуть повністю інкапсулювати мультиплексований потік системи DAB + в STL (посилання студія-передавач), значно знизивши витрати, пов'язані з дискретними каналами подачі[8].

DAB + мультиплексор або інфраструктура 'head-end' складається з

- один ансамблевий мультиплексор
- один ансамблевий контролер для розподілу потужностей студійних мультиплексорів та забезпечення функціональних можливостей передачі, таких як сигналізація, синхросигнал та побудова потоку ETI[8]
- один або кілька студійних мультиплексорів
- один або більше пов'язаних студійних мультиплексних контролерів, які визначають та керують наданими послугами та вмістом, включаючи тип звуку та швидкість передачі даних, введення пов'язаних з програмою введення даних, такі як прокрутка тексту (сегмент динамічної мітки / DLS) та зображень (слайд-шоу / SLS), а також даних послуги для функціональності, такі як електронні довідники програм[8]
- один або кілька кодувальників звуку на студію або ансамбль мультиплексорів

Студійні мультиплексори, як правило, функціонують незалежно один від одного різними радіомережами.

Кожен студійний або службовий мультиплексор, як правило, також має пов'язаний мультиплексор даних або сервер даних для надання пов'язаних з програмою даних (PAD) та послуг передачі даних[4].

Структура сімейства стандартів DAB + пропонує широкий спектр архітектури, що дозволяє оптимізувати та мінімізувати витрати для всіх типів комбінацій та вимог мережі, від ансамблів, що управляються єдиною мережею з декількома каналами, наприклад. BBC у Великобританії, до ансамблів, які мають 3 або більше незалежних комерційних мереж, у кожній з яких є декілька послуг, або до комбінації публічних, приватних і навіть широких громадських служб на спільному мультиплексі[8].

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

### 2.1.3 Стандарт Integrated Services Digital Broadcasting (ISDB)

Японський стандарт, що використовується для радіо та телевізійного мовлення.

Різні варіації ISDB відрізняються, головним чином, типом використовуваних модуляцій через вимоги різних діапазонів частот. Діапазон ISDB-S 12 ГГц використовує модуляцію PSK, цифровий звуковий мовний діапазон 2,6 ГГц використовує CDM, а ISDB-T (у діапазоні УКХ та / або УВЧ) використовує COFDM з PSK/QAM[4].

Крім передачі аудіо та відео, ISDB також визначає з'єднання даних (Трансляція даних) з Інтернетом як канал повернення через декілька медіа (10Base-T / 100Base-T, Модем телефонної лінії, Мобільний телефон, Бездротова мережа (IEEE 802.11) тощо.) та з різними протоколами[4]. Цей компонент використовується, наприклад, для інтерактивних інтерфейсів, таких як трансляція даних (ARIB STD-B24), та електронних довідників програм (EPG) [4].

Специфікація ISDB описує багато мережевих інтерфейсів, але головне, загальний інтерфейс для системи умовного доступу (CAS). Хоча в ISDB є приклади впровадження різних типів систем CAS, в Японії використовується система CAS під назвою "B-CAS". ARIB STD-B25 визначає систему загального алгоритму скремблювання (CSA) під назвою MULTI2, необхідну для (де-) скремблювання телебачення[4].

Діапазон частот наземного телевізійного мовлення, який підходить для мобільного зв'язку, OFDM (мультиплексування ортогонального частотного поділу), який витримує перешкоди, спричинені кількома шляхами (затримка хвиль), метод модуляції, який підходить для зв'язку з мобільними телефонами та мобільними приймачами, потужна функція виправлення помилок, і т.д., були прийняті, щоб забезпечити хороший зв'язок з мобільними телефонами та мобільними приймачами.

Щодо технології стиснення інформації та технології мультиплексування, MPEG-2 було прийнято після розгляду різноманітних

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

сумісних комунікацій із цифровим мовленням (такими як наземне цифрове телевізійне мовлення, BS цифрове мовлення, CS цифрове мовлення). MPEG-2 пропонує загальну базу для обробки сигналів, що призводить до зниження виробничих витрат приймачів за допомогою LSI-мікросхеми та консолідації приймачів, а також легкого обміну даними з іншими носіями[8].

Оскільки ця система мовлення має загальну сегментну структуру з наземним цифровим телевізійним мовленням, приймачі можуть бути консолідовані.

MPEG-2 AAC (Advanced Audio Coding) та SBR (Spectral Band Replication) були прийняті[4]. Однак SBR необов'язковий.

Ця система задовольняє стандарту ITU-R, що забезпечує високоякісну багатоканальну каналізацію з низькою швидкістю передачі бітів 144 кбіт / с. Його прийняли цифрове мовлення BS та наземне цифрове телевізійне мовлення[8]. Прийняття до складу ДСБ було результатом розгляду крос-медіа-комунікацій.

Оскільки ця система мовлення має загальну сегментну структуру з наземним цифровим телевізійним мовленням, приймачі можуть бути консолідовані.

Для тарифікованого мовлення прийнята система кодування. Це система MULTI2, яка вже прийнята для наземного цифрового телевізійного мовлення, цифрового мовлення BS та цифрового мовлення CS[4]. Прийняття до складу ДСБ було результатом розгляду крос-медіа-комунікацій. Введення зарядженого мовлення залежить від результату після вивчення техніко-економічного обґрунтування[8].

## 2.2 Стандарти двостороннього зв'язку

Такий тип стандартів зв'язку використовується для встановлення двонаправленого зв'язку між абонентами, на відміну від стандартів, наведених в попередньому підрозділі. Основні сфери застосування – мобільний зв'язок та радіозв'язок (як цивільний, так і спеціальний). В данній

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

роботі розглядаються лише стандарти, що охоплюють мобільний стільниковий зв'язок.

### 2.2.1 Стандарт CDMA2000

Розроблений компанією 3GPP2 як зворотно-сумісний наступник набору стандартів cdmaOne другого покоління (IS-95).

Мета CDMA2000 - забезпечити шлях міграції від вихідної системи cdmaOne / IS-95 через формат CDMA2000 1X до подальших форматів високої швидкості[7]. Всі ці стандарти були стандартизовані у форматі IS, а схема шляху міграції наведена на рис. 2.1.

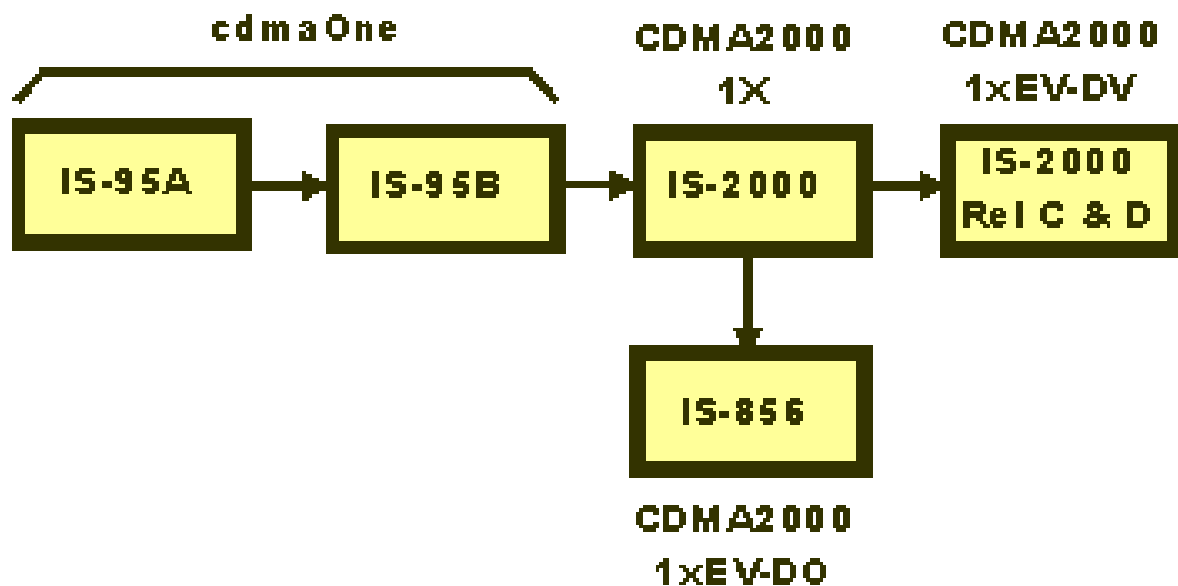


Рис. 2.1: схема міграції CDMA2000

Формат CDMA2000 1X є базовим стандартом 3G, але в тому, що називається CDMA2000 1xEV, є подальші розробки[9]. В основному є два маршрути еволюції, які були запропоновані спочатку, лише один був розгорнутий.

Перший з них, відомий під назвою CDMA2000 1xEV-DO, є чимось стороною від основної еволюційної розробки стандарту. Він визначений під IS-856, а не IS-2000, і як зазначено в назві, він переносить лише дані, але зі швидкістю до 3,1 Мбіт / с у прямому напрямку та 1,8 Мбіт / с у зворотному напрямку, швидкість у зворотному ланцюзі модернізується як частина випуску А стандарту[9]. Перша комерційна мережа CDMA2000 1xEV-DO була розгорнута компанією SK Telecom (Корея) у січні 2002 року.

Другий - CDMA2000 1X EV-DV (Evolution Data and Voice). Ідея полягала в тому, що ця система здійснюватиме як передачу даних, так і голосові послуги[9]. Вона ніколи не була розгорнута, оскільки система EV-DO була розгорнута і не було потреби в послугі передачі даних і голосу, оскільки голос можна переносити на DO як VoIP, так і повертаючись до формату CDMA2000 1X[7].

Терміни CDMA2000 1XRTT і 3XRTT відносяться до того, що називають "технологіями радіопередачі". Оригінальний IS-95 та розгортання CDMA2000 використовували інтервал каналів 1,25 МГц[9]. Це забезпечило фактично першу фазу розробки та впровадження 3G. Однак для підвищення продуктивності, де можливо, використовуючи такі технології, як 1xEV-DO і 1xEV-DV, смуга пропускання каналу в 1,25 МГц вважалася недостатньою для навіть більш високих швидкостей передачі даних[9]. Відповідно, збільшуючи пропускну здатність, були можливі більш високі швидкості передачі даних. Подальша еволюція системи CDMA2000 передбачає використання пропускну здатності каналів у 3 рази більше стандартної смуги пропускання 1,25 МГц під тим, що називалося 3XRTT[9]. Подальше пропускну здатність збільшується до 5X, 7X і так далі.

Для технології CDMA2000 використовувалась 1XRTT. Швидкість розповсюдження 1 (SR1), де сигнал розповсюджувався на пропускну здатність 1,25 МГц[7]. Тут швидкість розповсюдження була такою ж, як і для IS-95, тобто 1,22288 мкс[9]. Для технології 3XRTT було використано коефіцієнт розповсюдження 3 (SR3)[7]. Тут коефіцієнт розповсюдження становив 3,6864 мкс. Було встановлено, що якщо швидкість розповсюдження залишається такою ж, але швидкість передачі даних збільшується, як це відбувається при завантаженні відео та інших програмах 3G, посилення обробки зменшується. Відповідно, потрібно вдосконалити охоплення та потужність сигналу, щоб відповідати новим умовам. Збільшуючи швидкість розповсюдження, можна збільшити продуктивність без необхідності вдосконалення покриття[9].

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

Для покращення продуктивності CDMA2000 1X, IS2000 над cdmaOne IS-95 було введено ряд оновлень та змін. Однак у всіх випадках підтримується зворотна сумісність, що дозволяє і мобільним пристроям IS-95, і CDMA2000 отримувати доступ до одних і тих же базових станцій[9]. Це забезпечило економічно ефективний шлях оновлення як для користувачів, так і для операторів.

Для CDMA2000 1X було використано кілька нових методів кодування та розповсюдження, що дозволило досягти значно більших можливостей[7].

Перша основна зміна CDMA2000 1X полягала в тому, що використовувані коди Уолша були збільшені з 64 біт для IS-95 до 128 біт для CDMA2000 1X[9]. На додаток до цього, CDMA2000 1X також використовував більше функцій кодування помилок і використовував турбокоди, а не згорткові коди, що використовуються для IS-95[9]. Це дозволило надсилати дані з більшою швидкістю. На додаток до цього переплетення та повторення символів використовувались для забезпечення різних швидкостей передачі даних.

Турбокоди були введені в CDMA2000 1X. Вони були новим класом кодів виправлення помилок, які дозволили швидкості передачі через галасливий канал наблизитись до межі "Шеннона". Принцип турбокодування перше був запропонований в 1993 році професорами Клодом Берро та Аленом Гласьє. Спочатку до їхніх тверджень про те, що коди могли б подвоїти пропускну здатність для даної потужності, ставились скептично, але їх висновки в кінцевому підсумку виявилися правдивими. Турбо-кодери використовують потужні перемежувачі, що зменшують сприйнятливність потоку даних до випадкового та імпульсивного шуму. Працюючи над "м'якими" бітами радіоприймача, турбокоди дозволяють декодеру витягувати максимальний рівень даних із зашумлених сигналів. Для турбокодів потрібні два кодери та два декодери[7]. Ці блоки працюють паралельно і працюють синергетично.

Окрім покращень у розповсюдженні та генеруванні каналів, відбулися також зміни у самому повітряному інтерфейсі. Прямі посилання IS-95 використовувала форму QPSK, де дані обох каналів I та Q однакові[9]. Однак для CDMA2000 1X канали I та Q були різними, і це дало перевагу в тому, що половина пропускної здатності може бути використана для тієї ж кількості мікросхем, або вдвічі більше мікросхем може бути надіслано в одній смузі пропускання[7]. Хоча це робило прийом більш чутливим до фазових помилок, інші вдосконалення включали вдосконалену систему управління потужністю вперед та різноманітність передачі вперед.

### 2.2.2 Стандарт UMTS-TDD

UMTS-TDD, аббревіатура для Універсальної мобільної телекомунікаційної системи (UMTS) - дуплексизації з поділом за часом (TDD), є стандартизованою версією мереж UMTS, що використовують UTRA-TDD, 3GPP. UTRA-TDD - це UTRA, яка використовує дуплексне ділення за часом поділу для дуплексування. [7] Незважаючи на повну реалізацію UMTS, вона в основному використовується для забезпечення доступу до Інтернету в обставинах, подібних до тих, де може використовуватися WiMAX[9]. UMTS-TDD не є безпосередньо сумісним з UMTS-FDD: пристрій, призначений для використання одного стандарту, не може, якщо спеціально не розроблено для роботи іншого, через різницю технологій повітряного інтерфейсу та частот[9]. Це формальніше звучить як IMT-2000 CDMA-TDD або IMT 2000 Time-Division (IMT-TD).

Два UMTS повітряні інтерфейси (UTRA) для UMTS-TDD - це TD-CDMA та TD-SCDMA[9]. Обидва інтерфейси використовують комбінацію двох методів доступу до каналів, множинний доступ до поділу коду (CDMA) та множинний доступ з поділом за часом (TDMA): смуга частот поділяється на часові інтервали (TDMA), які далі поділяються на канали за допомогою кодів розповсюдження CDMA[9]. Ці інтерфейси класифікуються як TDD, оскільки часові інтервали можуть бути розподілені як для висхідної, так і для низхідній лінії зв'язку.

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24



Дуплексна схема передбачає спосіб організації передавача і приймача, щоб вони могли передавати і приймати. Існує кілька методів, які можна застосувати. Для додатків, включаючи бездротові та стільникові телекомунікації, де потрібно, щоб передавач і приймач могли працювати одночасно, застосовуються дві схеми. Один з відомих як FDD або дуплексний поділ частоти використовує два канали, один для передачі, а другий для приймача[9]. Інша схема, відома як TDD, дуплексний поділ часу використовує одну частоту, але виділяє різні часові проміжки для передачі та прийому[9].

При використанні системи TDD існує ряд характеристик, що стосуються систем TDD. Ці характеристики необхідно враховувати при розробці або використанні систем TDD.

Використання непарних смуг: зазвичай в трафіку низхідній лінії (мережа до мобільного) більше трафіку, ніж у висхідній лінії зв'язку (мобільний до мережі)[7]. Відповідно, оператор може виділити більше часу передачі низхідній лінії зв'язку, ніж висхідній лінії зв'язку. Це неможливо при парному спектрі, необхідному для систем FDD, коли неможливо перерозподілити використання різних діапазонів[9]. В результаті цього можна дуже ефективно використовувати наявний спектр.

Переривчаста передача: у будь-якій системі TDD необхідно перемикатися між транзитом і отриманням[7]. Це займає певну кількість часу. Мало того, що потрібен час, щоб мобільна та базова станція змінювались між передачею та прийманням з точки зору нарощування потужності вгору або вниз, разом з відстоюванням будь-яких перехідних процесів[9]. На додаток до цього потрібен час між передачею та прийманням, щоб пристосувати час передачі між мобільною та базовою станцією. В результаті потрібна охоронна стрічка.

Перешкоди для висхідної / низхідній лінії зв'язку: оскільки і висхідна, і низхідна лінії передають один і той же канал, між двома напрямками передачі можуть бути перешкоди. Щоб подолати це, базові станції

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

синхронізуються, щоб гарантувати, що вони не передають, коли приймається сусідня базова станція, інакше краще розміщення та можливий більший рівень потужності спричинить перешкоди[7].

Еквівалентні умови для висхідної та низхідної лінії зв'язку: оскільки і висхідна, і низхідна лінія використовують один і той же канал, вони підпадають під однакові умови поширення[7]. У системах FDD, що використовують різні частоти для висхідної та низхідній лінії зв'язку, є значні відмінності[9]. Використовуючи однакову частоту затухання сигналу, можна протидіяти більш ефективно.

### 2.2.3 Стандарт WCDMA

WCDMA (Широкосмуговий множинний доступ з кодовим розділенням), також називається UMTS (універсальна мобільна телекомунікаційна система) - стандарт, що використовується в мережах мобільного зв'язку 3G. Він підтримує звичайні стільникові голосові, текстові та MMS- послуги, але також може передавати дані з високою швидкістю, що дозволяє мобільним операторам надавати додатки з більшою пропускну здатністю, включаючи потоковий та широкосмуговий доступ до Інтернету[10].

WCDMA використовує метод доступу до каналу DS-CDMA з парою 5 МГц каналів[9]. На противагу цьому, конкуруюча система CDMA2000 використовує один або більше доступних каналів 1,25 МГц для кожного напрямку зв'язку[9]. Системи WCDMA широко піддаються критиці за їх широке використання спектру, що затримує розгортання в країнах, які відносно повільно діють при виділенні нових частот спеціально для послуг 3G.

Конкретні частотні смуги, спочатку визначені в стандарті UMTS є 1885-2025 МГц для передачі до базової станції (висхідній лінії зв'язку) і 2110-2200 МГц для базової станції до мобільної станції (низхідній лінії зв'язку)[10]. У США замість цього використовують 1710–1755 МГц і 2110–2155 МГц, оскільки діапазон 1900 МГц вже використовувався[10]. Хоча UMTS2100 є найбільш широко розгорнутим діапазоном UMTS, деякі оператори UMTS

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

деяких країн використовують діапазони 850 МГц (900 МГц в Європі) та / або 1900 МГц (незалежно, тобто висхідна лінія та низхідна лінія знаходяться в одній смузі), зокрема в США на AT & T Mobility , Новій Зеландії Telecom Нової Зеландії по мобільній мережі XT і в Австралії Telstra на Next G мережі[9]. Деякі провайдери, такі як T-Mobile використовують номери смуг для ідентифікації частот WCDMA. Наприклад, діапазон I (2100 МГц), діапазон IV (1700/2100 МГц) та діапазон V (850 МГц) [9].

Загальна схема організації WCDMA наведена на рисунку 2.2, де частота чипів (Chip rate)  $C_r = 3,84$  МГц, частота квантування ЦАП-АЦП = 15.36 МГц, швидкість передачі даних (Symbol rate)  $= C_r / S_f$  , коефіцієнт розсіювання (spread factor)  $S_f = 4, \dots, 1920$  [10].

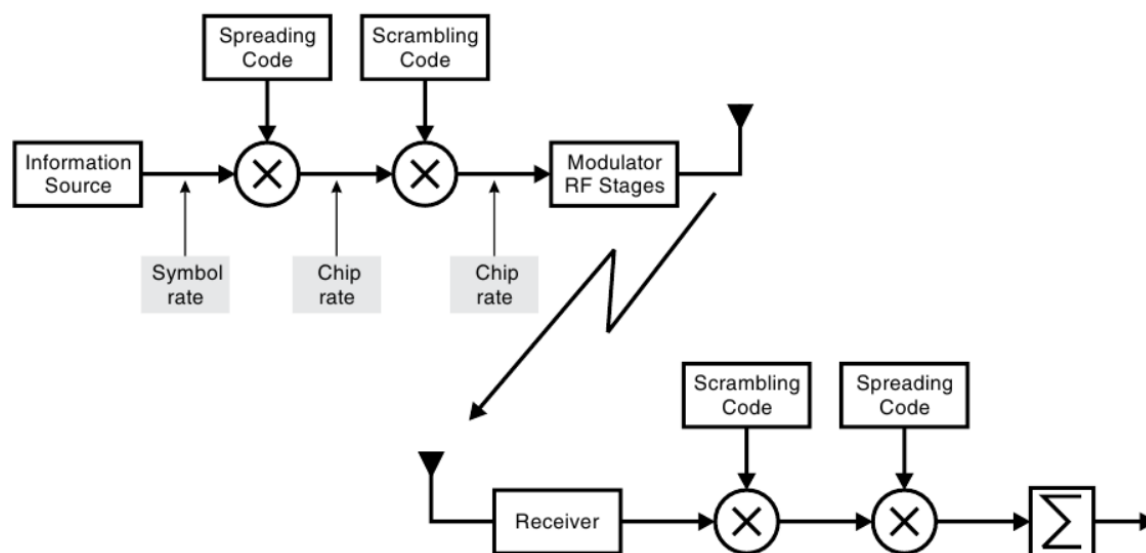


Рис. 2.2: загальна схема організації WCDMA

В данному стандарті наявний фільтр – квантувальць, що встановлений і в передавачі, і в приймачі, що подавляє побічні пелюстки (наведені на рис. 2.3) без додавання шуму[10].

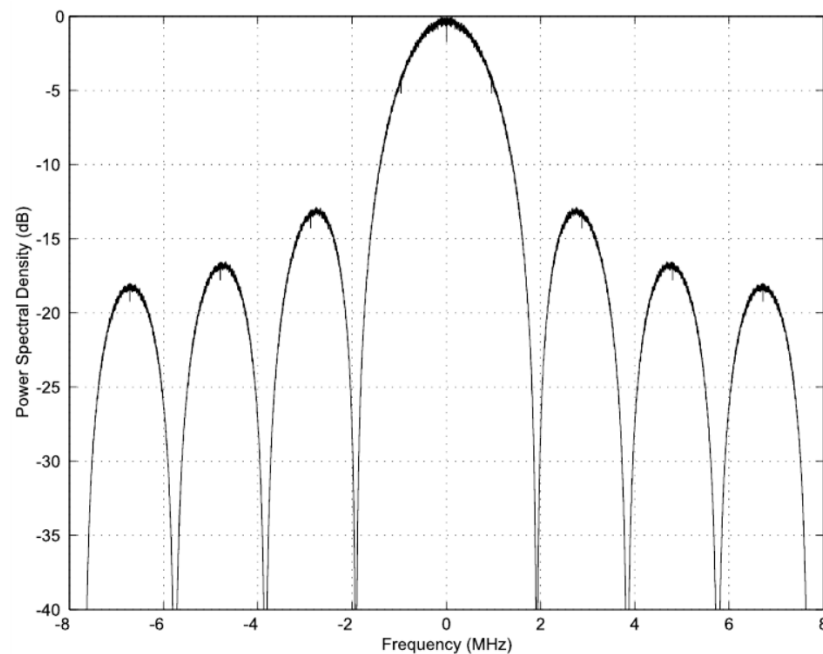


Рис. 2.3: спектр WCDMA

У WCDMA наявні наступні канали:

- фізичний канал загальних пакетів - Physical common packet channel (PCPCH) [10]
- Канал індикації стану - CPCH status indication channel (CSICH) [10]
- канал детектування колізії або призначення каналу - collision detection / channel assignment indicator channel (CD / CAICH) [10].
- канал зворотного управління - downlink physical control channel (DL-DPCCH) [10]

Управляє потужністю, знаходиться в першому слоті

- первинний канал загального управління фізичного каналу - primary common control physical channel (PCCPCH) [10]

Управляє інформацією про передані дані - в 1 слоті 18 біт даних у 2304 чіпах, а в перших 256 чіпах - канал відключений, щоб передавати синхронізуючі канали:

- первинний і вторинний синхронуючі канали - primary and secondary synchronisation channels (P-SCH, S-SCH) [10]

Передається в перших 256 чіпах слота. Закодовані в 256 чіпах ієрархічної послідовністю Голі. Застосовуються, щоб визначити наявність станції. S-SCH крім того, задає групу кодів даної станції.

- первинний загальний канал пілотного сигналу - primary common pilot channel (P-CPICH). [10]

Передається набір спеціальних символів безперервно всередині слота і протягом 15 слотів вікна. Для встановлення фази станції. Необхідний для демодуляції інших каналів.

- вторинний загальний канал пілотного сигналу - secondary common pilot channel (S-CPICH). [10]

Передається набір спеціальних символів безперервно всередині слота і протягом 15 слотів вікна. Для уточнення фази станції для адаптивних антен.

- вторинний канал загального управління - Secondary common control physical channel (S-CCPCH), [10]

Передає транспортні канали - прямого доступу - forward access channel (FACH) і сторінковий канал - paging channel (PCH). Один слот каналу включає код TFCH, дані і пілотний сигнал.

- канал індикації підтвердження прийому – AICH[10]

Передає +1 (ACK), - 1 (NACK) і 0 як NULL протягом 20 мс - в  $33 * 15$  примірниках. Дані передаються одним з 16 ортогональних векторів, кожен довжиною 32 символу. Ці вектори - коди Адамара парного порядку, модулюються і передаються з уповільненням 256.

- канал загальних даних - Physical downlink shared channel (PDSCH). [10]

Крім того, даній системі застосовано протокол RRC, що являється головним управляючим протоколом[9]. Архитектура протоколу RRC проілюстрована на рисунку 2.4. Основними функціями архітектури є виділений функціональний об'єкт управління (DCFE), функціональний об'єкт підкачки та повідомлення (PNFE) та функціональний об'єкт управління широкомовної інформації (BCFE).

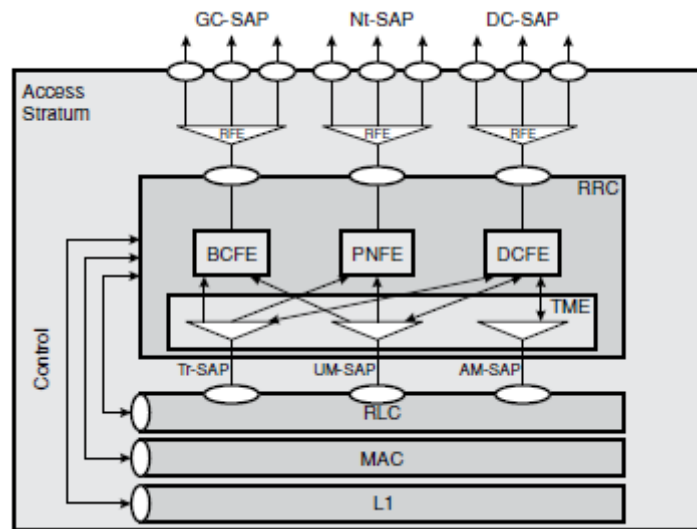


Рис. 2.4: архітектура протоколу RRC

Повідомлення RRC передаються між UE та WCDMA[10]. Вони використовуються для налаштування та управління з'єднанням RRC між UE та WCDMA. Повідомлення RRC можна легко згрупувати у чотири категорії: повідомлення управління зв'язком RRC; Повідомлення управління RB; Повідомлення про мобільність з'єднання RRC та повідомлення про вимірювання RRC[10].

Перша категорія повідомлень - це повідомлення управління RRC-з'єднаннями, які відповідають за встановлення та підтримку RRC-з'єднання в будь-якій формі. Повідомлення включають повідомлення про з'єднання RRC, повідомлення контролю безпеки та повідомлення системної трансляції інформації, а також повідомлення для передачі даних NAS[10].

SIB - це системна інформація, яка передається від UTRAN до UE. UE потрібно знайти та прочитати системну інформацію перед встановленням будь-якого радіозв'язку з UTRAN[9]. Однією з проблем розробки, пов'язаних із SIB, є те, що інформація надходить у різних видах. Деяка інформація оновлюється часто (наприклад, оцінки рівнів перешкод висхідній лінії зв'язку, виміряні у Вузлі В), а деяка інформація не потребує регулярного оновлення (наприклад, ідентифікатори комірок та системи). Крім того, повідомлення можуть бути довгими, а також різної довжини[9]. Структура

каналів широкомовної розробки призначена таким чином, щоб впоратися з цими різними обмеженнями.

Шифрування в данній системі реалізовано за допомогою скремблювання, тобто введення в систему скремблера, що зворотно шифрує послідовність бітів в вихідному масиві таким чином, що ймовірність появи нуля або одиниці в отриманному потоці не залежить від початкових даних.

Скремблювання в данній системі виконується в 3 етапи.

- На 1 етапі:

Якщо 2 сусідніх чіпа однакові, то для першого чіпа буде поворот на  $+45^\circ$ , а для другого - на  $-45^\circ$ . Це виходить множенням коду каналу на  $(1 + j(-1)^i)$ , де  $i$  - номер чіпа. В результаті - фаза сигналу змінюється на  $+45^\circ$  і  $-45^\circ$  на кожному чіпі[10].

- На 2 етапі:

Виконується множення на псевдовипадкову послідовність типу коду Голда. Оскільки ця послідовність реальна, то фаза змінюється на  $0^\circ$  або на  $180^\circ$  в залежності від цифри коду. Для цього на скремблирується код множиться сигнал в I-компоненті і в Q-компоненті (на  $+1$  або  $-1$ ) [10].

Так що, на 1-му етапі зміна фази виходить на  $90^\circ$  для пари чіпів, а на 2-му етапі - на  $0^\circ$  або  $180^\circ$ . Резульгуюче зміна фази -  $90^\circ$  або  $270^\circ$  ( $270^\circ$  - те ж саме, що  $-90^\circ$ ). А якщо зсув фази на  $90^\circ$  або  $-90^\circ$ , то кінець вектора не перетинає нульову точку[10].

- На 3 етапі:

Реальні складові коду дециміруються в часі на 2, тобто береться кожен 2-й відлік. Це можливо на підставі того, що код залишається той же самий в сусідніх чіпах. Така децимація впливає тільки на складову Q, в результаті чого фаза повертається на  $+45^\circ$  або  $-45^\circ$ [10]. Завдяки цьому, потужність складової I і складовою Q в середньому, однакові.

Для скрамблювання застосовуються короткі та довгі послідовності. Вони визначаються сигналами  $C_{short1}$ ,  $C_{short2}$  и  $C_{long1}$ ,  $C_{long2}$

$C_{\text{long1}}$  – реальна послідовність довжини  $2^{25}-1$ ,  $C_{\text{long2}}$  – та сама послідовність, але зі ссувом на 16777232 чипів[10].

$C_{\text{short1}}$  и  $C_{\text{short2}}$  - дуже довгі коди Касамі. Генератор такого коду - четверичной як на рис[10]. З цифр коду формується пара бінарних кодів. Застосовуються при високій інтерференції.

Виходячи з вище сказаного, комплексні коди скремблювання мають вигляд:

$$C_{\text{long}}(i) = C_{\text{long1}}(i) \{1 + j(-1)^i C_{\text{long2}}[2(i/2)]\}$$

$$C_{\text{short}}(i) = C_{\text{short1}}(i \bmod 256) \{1 + j(-1)^i C_{\text{short2}}[(i \bmod 256)/2]\}$$

Структура генератора короткого скремблюючого коду наведена на рисунку 2.5.

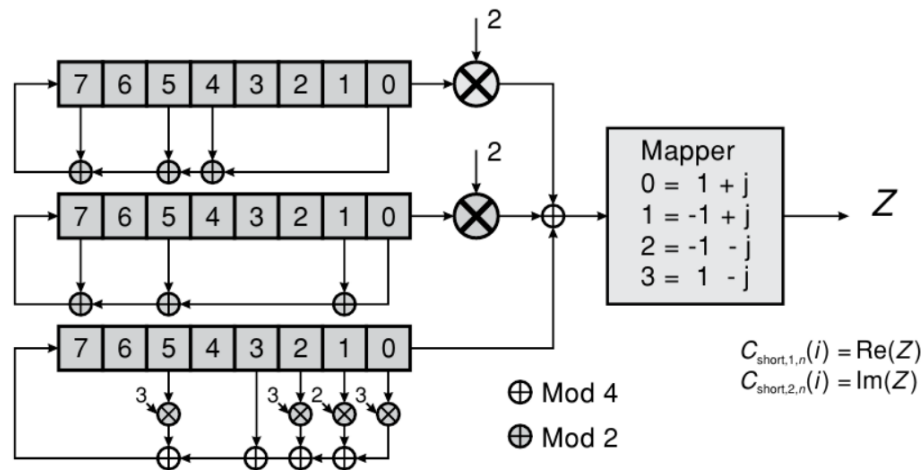


Рис. 2.5: генератор короткого скремблюючого коду

Якщо повідомлення надходять з періодом 10 мс, генератор скремблюючого коду скидається на межі 10 мс інтервалу[10].



## ВИСНОВКИ ДО РОЗДІЛУ 2

1. З огляду на особливості апаратної частини системи, а саме розташування приймача та передавача в межах однієї плати, доцільно реалізовувати саме стандарт двостороннього зв'язку, так як об'єктивних перешкод для його реалізації немає.
2. Виходячи з особливостей приведених вище стандартів, для розробки було обрано стандарт WCDMA.

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

### РОЗДІЛ 3

## ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ СИСТЕМИ

Для розробки програмного рішення було обрано мову VHDL. Розробка ведеться за допомогою середовища Aldec Active-HDL, з огляду на те, що у данному середовищі наявні методи симуляції роботи пристрою.

### 3.1 Опис ПЛІС та її основних компонентів

В структурі плати наявні наступні інтерфейси:

- Інтерфейс до цифрового інтерфейсу IQ LimeLight™ Digital LQ7002 в режимі подвійної швидкості передачі даних TRXIQ;
- Порт передачі даних у режимі реального часу між ПК та мікросхемою LMS7002.
- Підключення до FX3 Slave FIFO інтерфейс для передачі даних через USB3.0.
- Синхронізація TX-зразків із часовою міткою зразків RX;
- З'єднання SPI між мікросхемою LMS7002 та іншими бортовими пристроями;
- Плеєр WFM, який дозволяє завантажувати форму хвилі у зовнішню пам'ять DDR2 з хоста USB3.0 і переводити на інтерфейс LMS7002 RXIQ
- Реконфігурувані блоки PLL для синхронізації LMS7002.
- Внутрішні регістри SPI для управління FPGA.

Cyclone IV FPGA забезпечує інтерфейс GPIF II з контролером FX3 USB3.0. Існують дві кінцеві точки (EP0F та EP8F), реалізовані для контрольних даних, і дві кінцеві точки для потокових даних (EP01 та EP81). Кінцеві точки управління підключені до програмного забезпечення NIOS II, який забезпечує комунікаційні інтерфейси SPI та I2C для мікросхеми LMS7002M, TCXO DAC, фазового детектора ADF4002, датчика температури LM75. NIOS також забезпечує доступ до внутрішніх регістрів конфігурації SPI. Кінцеві точки потоку призначені для прийому та передачі IQ даних з / в LMS7002M.

Рис. 3.1 містить блок-схему з основними модулями.

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

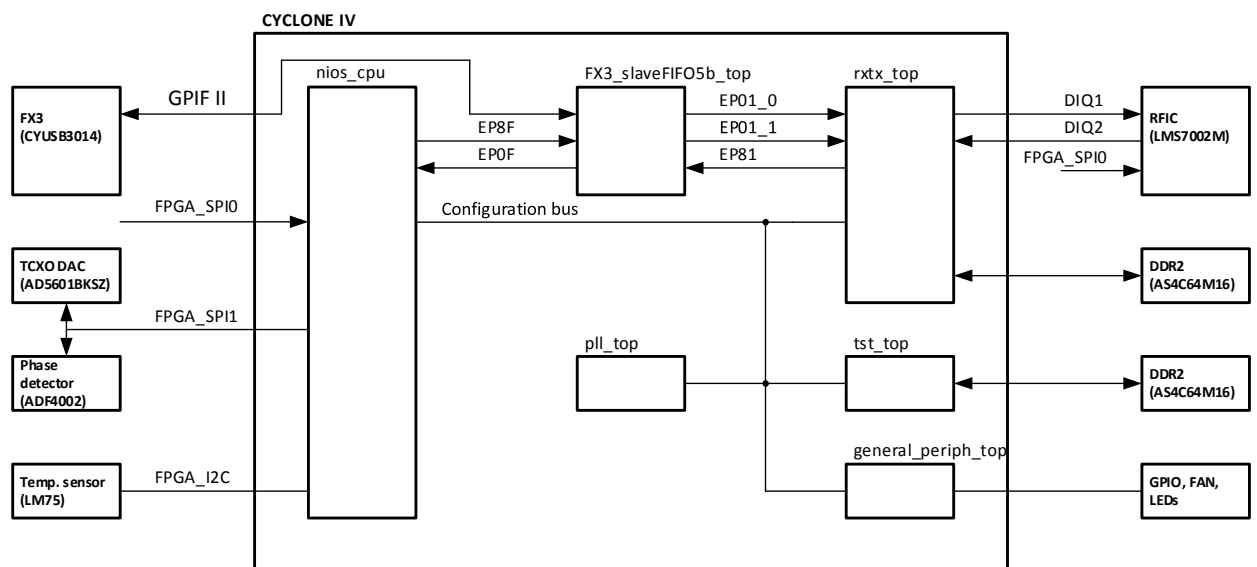


Рис. 3.1: блок-схема основних модулів

Основні модулі плати:

- nios\_cpu

NIOS II програмний процесор з регістрами пам'яті. Забезпечує контроль периферії[11].

- FX3\_slaveFIFO5b\_top

Забезпечує передачу даних між зовнішнім периферійним контролером USB 3.0 FX3 SuperSpeed та FPGA[12].

- rtx\_top

Отримує та передає логіку між FPGA та зовнішнім приймачем LMS7002[11].

- General\_periph\_top

Модуль управління для вбудованої периферії, такий як світлодіоди, GPIO, FAN.

- pll\_top

Забезпечує необхідні годинники для модуля rtx\_top.

- tst\_top

Логіка перевірки плати для перевірки зовнішньої пам'яті DDR2 та зовнішніх годинників[11].

На рисунку 3.2 показаний потік даних між основними модулями та схемою синхронізації.

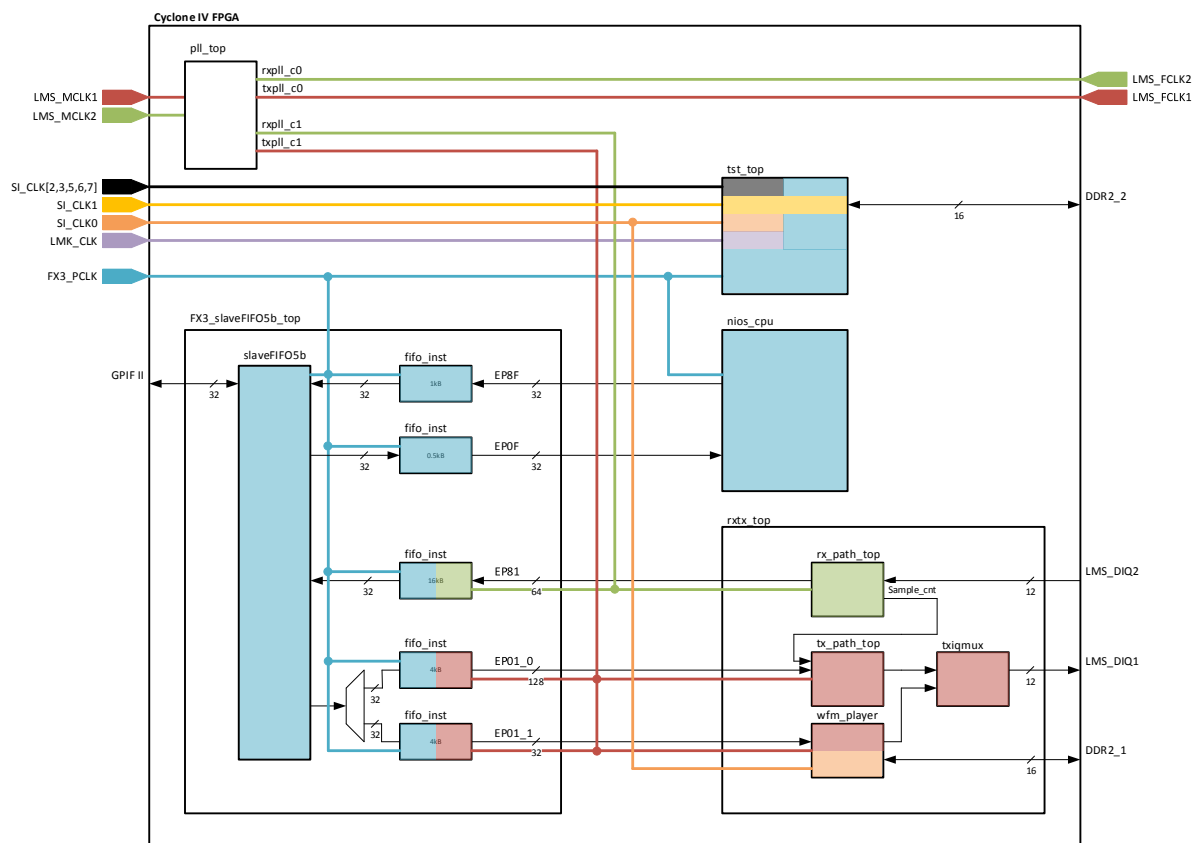


Рис. 3.2: потік даних між основними модулями та схемою синхронізації

На рисунку 3.3 показана блок-схема модуля `nios_cpu`. Цей модуль містить програмний процесор ALTERA NIOS II та доступні користувачу регістри конфігурації для інших модулів. Детальний опис модуля наведено нижче.

- `lms_ctr`

Екземпляр програмного забезпечення NIOS II. Процесор постійно контролює вхідний буфер FIFO, підключений до портів `exfifo_if`, і зчитує один пакет, що містить 64 байти. Процесор NIOS Виконує отриману команду і записує 64-байтний пакет відповідей у буфер FIFO, підключений до портів `exfifo_of[11]`.

- `cfg_top`

Модуль обгортки для регістрів конфігурації SPI.

- `fpgacfg`

Загальна конфігурація адресних регістрів 32x16b. Діапазон адрес 0x0000 - 0x001F

- `pllcfg`

Регістри конфігурації PLL. Діапазон адрес 0x0020 - 0x003F.

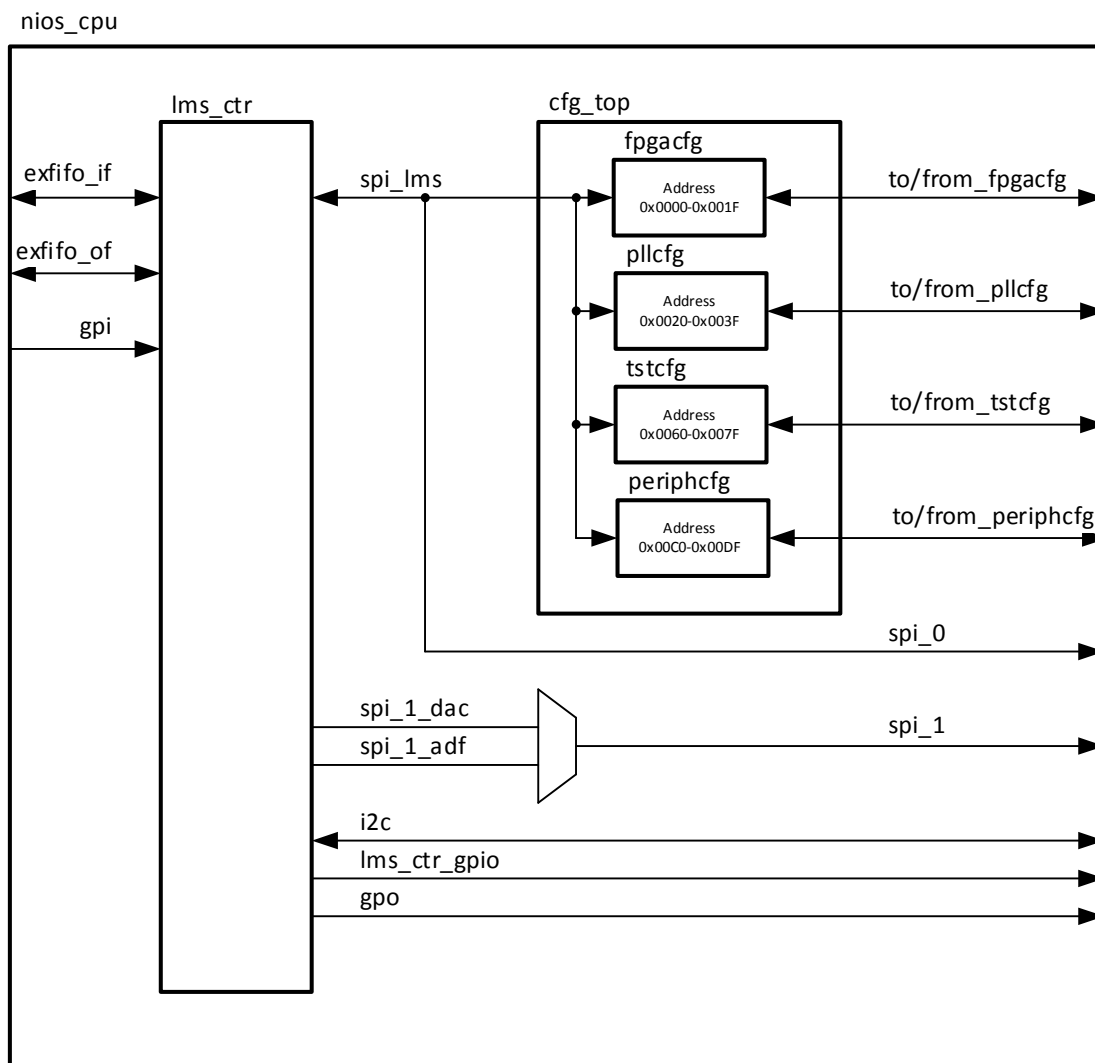


Рис. 3.3: блок-схема модуля nios\_cpu

- **tstcfg**

Реєструє конфігурацію модуля тестування. Діапазон адрес 0x0060 - 0x007F

- **periphcfg**

Регістри конфігурації периферії. Діапазон адрес 0x0020 - 0x003F.

FX3 Slave FIFO інтерфейс - FX3\_slaveFIFO5b

Забезпечує передачу даних між зовнішнім периферійним контролером USB 3.0 FS3 SuperSpeed і FPGA через інтерфейс GPIF II[11].

Весь обмін даними між модулем slaveFIFO5b та іншою логікою FPGA здійснюється через буфери FIFO. Модуль slaveFIFO5b постійно відстежує прапори статусу GPIF II та всі буфери FIFO[12]. Наприклад, внутрішня логіка записує IQ поточкові пакети, що містять дані 4kB, до буфера FIFO через порти EP81. Після того, як ведений модуль FIFO5b виявить, що буфери EP81 FIFO

містять дані 4 кБ, а графіки IIFlags вказують, що FX3 контролер готовий, всі дані зчитуються з буфера FIFO і записуються в контролер FX3 через інтерфейс GPIF[13].

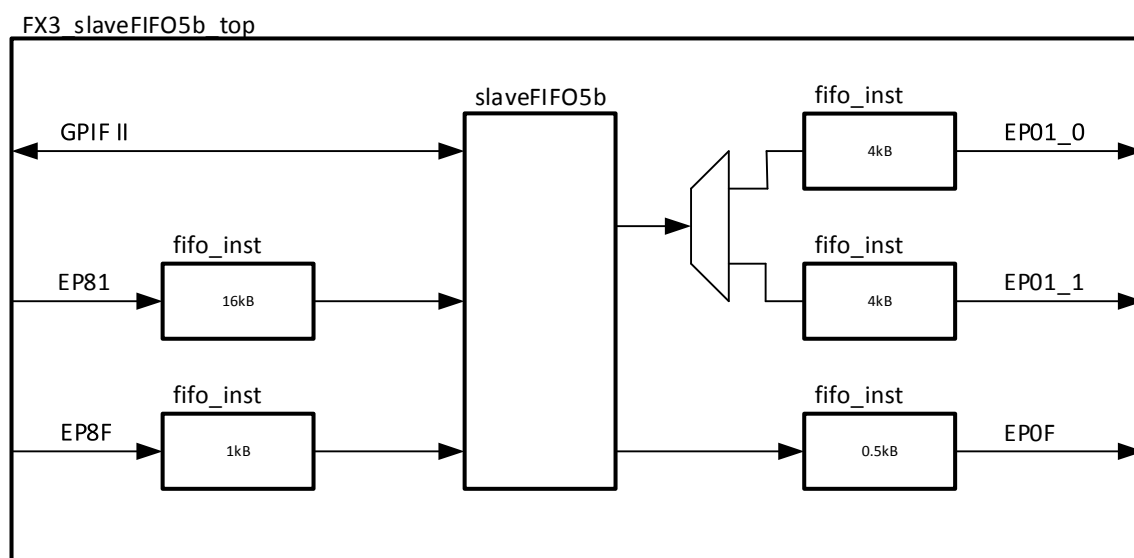


Рис. 3.4: блок-схема FX3\_slaveFIFO5b

Основні компоненти даного блоку:

- SlaveFIFO5b Забезпечує передачу даних між інтерфейсом GPIF II та внутрішніми буферами FIFO.
- fifo\_inst (EP81) Буфер FIFO кінцевої точки потоку розміром 16 кБ.
- fifo\_inst (EP01\_0) Буфер FIFO кінцевої точки потоку розміром 4 кБ.
- fifo\_inst (EP01\_1) Буфер FIFO кінцевої точки потоку розміром 4 кБ.
- fifo\_inst (EP8F) Контрольний буфер FIFO кінцевої точки розміром 1 кБ.
- fifo\_inst (EP0F) Контрольний буфер FIFO кінцевої точки розміром 0,5 кБ.

Основна функція модуля gctx\_top полягає в прийомі та передачі зразків IQ з / в мікросхему LMS7002 і забезпечує синхронізацію зразків IQ[13]. Блок-схема даного модуля надана на рис. 3.5.

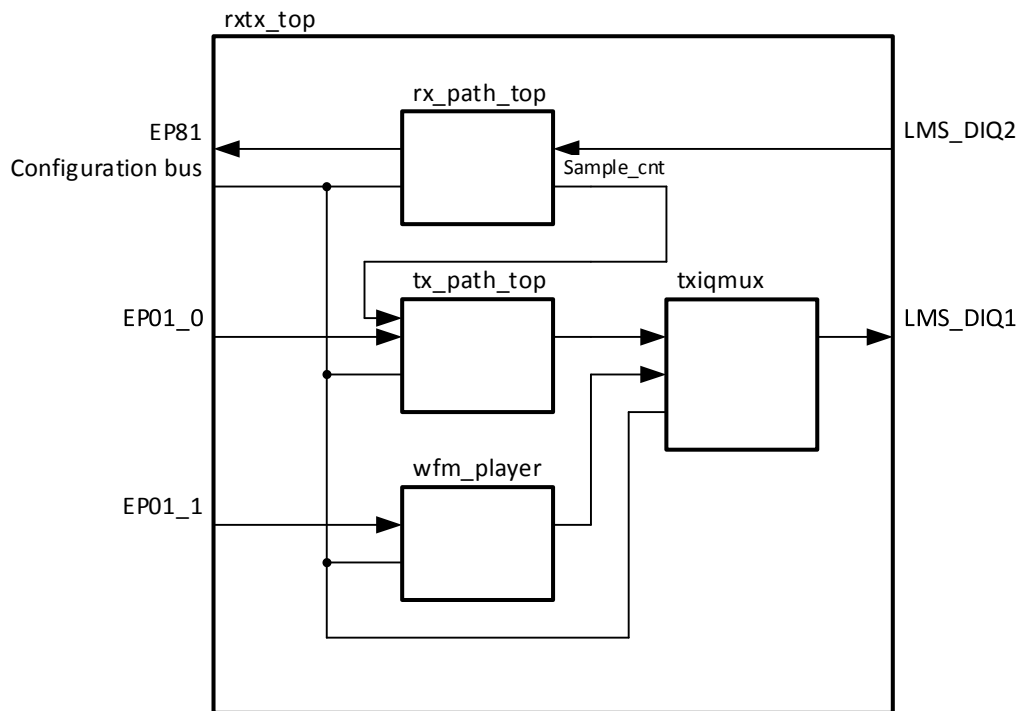


Рис. 3.5

Інтерфейс прийому - rx\_path\_top управляє модулями diq2fifo та data2packets, які безперервно пакують зразки IQ в пакети 4kB.

Пакети записуються в буфер EPIF FIFO 16kB EP81, щоб підтримувати постійний потік даних в короткі періоди, коли хост USB3.0 не може приймати дані[11]. Якщо хост USB3.0 зупиняє передачу даних на довший проміжок часу і чотири пакети буферуються в буфер 16 kB, виникає повна умова FIFO, а інші пакети відміняються. Коли хост починає отримувати дані після повного стану FIFO, хост повинен очікувати отримання цих чотирьох буферизованих пакетів[14].

Модуль rx\_path\_top забезпечує два 64-бітових лічильника вибірки. Єдина для логіки TX - tx\_path\_top. Логіка TX використовує цей лічильник для синхронізації переданих зразків LMS\_DQ1 з отриманими зразками LMS\_DIQ2[6]. Інший використовується для упаковки зразків LMS\_DI2 у 4 kB.

Коли увімкнено rx\_path\_top, модуль diq2fifo починає збирати зразки IQ з шини LMS\_DIQ2, зібрані зразки записуються в буфер FIFO, і кожне записування дозволяє smpl\_cnt: inst4 модуль збільшити його лічильне

значення[11]. Це означає, що значення лічильника збільшується в тій же безперервній швидкості, що і частота вибірки IQ.

Модуль `smpl_cnt: inst3` використовується для упаковки зразків `LMS_DI2` у 4 кБ[6]. Модуль `data2packets` зчитує зразки IQ під час всплеску з буфера FIFO, кожне зчитування дозволяє `smpl_cnt: inst3` модуль збільшувати свою лічильну величину. Одне зчитування, яке зчитується, заповнює один пакет від 4 КБ, і між цими пакетами є кілька непрацюючих циклів[11].

Модуль передачі `tx_path_top` читає зразки IQ з буфера `EP01_0` FIFO, упакованого в пакети 4 кБ. Заголовок пакета (див. Документ протоколу потоку) містить номер вибірки (або так звану часову марку), за якою пакет повинен передаватися[12].

Використовуючи номери вибірки з `rx_path_top` та отримані номери вибірки в заголовку пакету, передані зразки IQ можуть бути синхронізовані з отриманими зразками IQ.

Модуль `p2d_wr_fsm` відокремлює заголовки пакету та корисне навантаження[14]. Корисне навантаження пакетів записується в один з чотирьох 4 кБ буферів FIFO, розташованих в модулі `packets2data`, а заголовок пакета зберігається в модулі `p2d_rd`[14]. Цей модуль може працювати в двох режимах:

- Синхронізація включена

Модуль порівнює отриманий номер вибірки із заголовка пакета та номер вибірки з `rx_path_top`. Коли номер вибірки з отриманого пакету дорівнює номеру вибірки модуля `rx_path_top` (це означає, що пора надіслати TX-пакет), починається процес зчитування і зразки IQ передаються на інтерфейс `LMS_DIQ1`[14]. Коли число вибірки з отриманого пакету більше, ніж номер вибірки модуля `rx_path_top` (це означає, що отриманий пакет повинен бути відправлений через деякий час) `p2d_rd` чекає, поки номер вибірки не буде рівним. Коли число вибірки від отриманого пакету менше, ніж номер вибірки модуля `rx_path_top` (це означає, що пакет прийшов до запізнення), відповідний буфер FIFO очищається[14].

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40



- Синхронізація відключена

Модуль не порівнює кількість вибірок, і кожен отриманий пакет передається в інтерфейс LMS\_DIQ1[14].

### 3.2 Опис алгоритмів та компонентів програми

Для реалізації обраного стандарту було розроблено програмне рішення. Розробка спиралась на кілька існуючих робіт, зі схожою тематикою [15] [16].

Для коректної роботи програми її було поділено на модулі, кожен з яких реалізує певну функцію, необхідну для відтворення роботи системи згідно зі стандартом WCDMA. Опис окремих модулів наведено нижче.

#### 3.2.1 Модуль `main_controller`

Основний керуючий модуль програми, реалізує передачу даних між спеціалізованими модулями, та виклик їх на виконання.

Для цього в нього імплементовано зв'язок з командним протоколом ПЛІС[14], завдяки чому даний модуль може посилати сигнал до початку роботи всім залежним модулям.

Крім того задачою данного модулю є передача службових перемінних до підпорядкованих модулів, що переключають їх у режим роботи при передачі чи при прийомі. Це необхідно для оптимізації роботи програми, а також для зменшення загальної довжини коду.

#### 3.2.2 Модуль `mixer`

Реалізує цифрове мікшування сигналу.

Мікшер працює від 325 МГц до 3,8 ГГц в межах специфікації даних. До цього процесу змішування сигнал розщеплюється і подається по двох розділених, але однакових шляхах. Цей процес створює вбудовані та квадратурні компоненти нашого сигналу за допомогою простого фазового обертання годинника змішувача. Ефективно це подвоює ефективну пропускну здатність провідника, оскільки синфазні та квадратурні сигнали є ортогональними (пропускна здатність - від  $f_s/2$  до  $f_s/2$ ). смуга смуг і шум. Комбінований підсилювач трансимпанції (TIA) та аналоговий фільтр налаштовані разом, щоб підтримувати потрібну пропускну здатність аналога,

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

яка може становити від 200 кГц до 20 МГц. ТІА діє як однополюсний фільтр, а аналоговий програмований фільтр - Баттерворт третього порядку.

### 3.2.3 Модуль modulator

Реалізує квадратурну модуляцію, використовувану в стандарті WCDMA. Для реалізації данного типу модуляції недостатньо просто проманіпулювати з даними, і доводиться використовувати декілька фільтрів.

Піднятий косинусний фільтр - це тип фільтра Найкіста. Піднятий косинусний фільтр визначається через величину альфа. Для WCDMA альфа приймає значення 0,22[10].

Дотримуйтесь антисиметричного спектру щодо його відсічення. Чим менше альфа, тим чіткіше відсікання в частотній області. Зауважте також, що імпульсні реакції всі проходять через нуль в одній і тій же точці - нульова міжсимвольна інтерференція. Зауважте також, що величина дзвінка у часовій області стає більшою, коли значення альфа стає меншим.

Фільтр RRC визначає функцію передачі фільтра, який повинен використовуватися в передавачі для системи WCDMA. Представлення часової області для RRC виражено за допомогою формули

$$h(t) = \frac{\sin[(1-\alpha)\pi\frac{t}{T}] + 4\alpha(\frac{t}{T})\cos[(1+\alpha)\pi\frac{t}{T}]}{(\pi\frac{t}{T})[1-(4\alpha\frac{t}{T})^2]} \quad (3.1)$$

Крім того, необхідна фільтрація на стороні приймача.

Фільтрація в приймачі необхідна для відхилення шуму та перешкод, наявних у приймачі. Дії фільтра передачі та фільтра прийому поєднуються разом. Саме комбіновані ефекти передавального фільтра і приймального фільтра повинні враховуватися і повинні мати властивості фільтра Найквіста. Розглянемо рисунок 3.26, на якому показано поєднання фільтру передачі та приймача. НТ (f) - функція передачі для передавального фільтра в частотній області. НР (f) - функція передачі фільтра прийому в частотній області. Комбінований ефект двох фільтрів задається множенням двох передавальних функцій, як

					ІА/ЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

$$H_{SYS}(f) = H_T(f)HR(f) \quad (3.2)$$

Модуль не враховує будь-які ефекти функції передачі частоти домену, запроваджені радіоканалом. Так як в приймачі використовується той же фільтр, що і в передавачі ( $HR(f) = HT(f)$ ); це те, що називається відповідним фільтром і має оптимальну продуктивність. Потім поєднана функція передачі передавача та приймача задається як:

$$H_{SYS}(f) = [HT(f)]^2 \quad (3.3)$$

Функція передачі передавача задається квадратним коренем комбінованої функції передачі. З попереднього обговорення нам потрібно, щоб комбінована функція передачі була підвищеним косинусом ( $H_{SYS}(f) = RC$ ).

$$HT(f) = \text{Sqrt}[H_{SYS}(f)] = \text{Sqrt}(RC) \quad (3.4)$$

Отже, з рівняння вище, передавальний фільтр повинен бути квадратним коренем піднятого косинусу або корінсом, що піднімається коренем (RRC).

### 3.2.4 Модуль scrambler

Даний модуль розраховує скрамблюючі коди, необхідні для шифрування інформації.

Ідеальним кодом скремблювання була б справді випадкова двійкова послідовність, але проблема такої схеми полягає в тому, щоб послідовність була доступна і в передавачі, і в приймачі. Рішення полягає у використанні псевдовипадкових послідовностей на основі деякого алгоритму, який дозволить створити псевдовипадкову послідовність як у передавачі, так і приймач. Цей тип послідовності також зазвичай називають псевдошумовою послідовністю (або послідовністю PN). Для досягнення своїх цілей код для скремблювання повинен мати певні властивості. Дві основні властивості називаються автокореляцією та перехресною кореляцією. Автокореляція - це статистична властивість коду скремблювання, який сам по собі стосується коду. Перехресна кореляція є властивістю двох взаємодіючих кодів скремблювання.

Коли еталонна послідовність і послідовність вільного запуску ідеально вирівнюються, на виході літа є піковий сигнал, який дорівнює кількості мікросхем у послідовностях, що множать і підсумовують. Коли дві послідовності не вирівняні, вихід менший за цей піковий рівень.

Перехресна кореляція схожа на автокореляцію, за винятком того, що еталонна послідовність і послідовність тестування різні. Перехресна кореляція - це міра відмінностей між двома послідовностями. Для таких систем, як WCDMA, ідеальним перехресним співвідношенням є те, що має низьке значення[10].

Проста PN послідовність на основі M-послідовностей показана на рисунку 3.22. Довжина послідовності становить 24 -1 біт.

- M-послідовності часто використовуються в системах CDMA для поширення та скремблювання.
- M-послідовності мають дуже хороші, добре визначені властивості автокореляції.
- M-послідовності не мають дуже хороших перехресних кореляційних властивостей.
- M-послідовності отримані з рівняння, яке називається примітивним многочленом.

M-послідовності, засновані на примітивних поліномах, мають такі характеристики. Довжина M-послідовності становить  $2^n - 1$ , де  $n$  - порядок примітивного полінома та кількість бітів у регістрі зрушень, що використовується для його створення. У повній послідовності  $2^n - 1$  біт число 1s та число 0s відрізняється щонайбільше 1. За винятком усіх кодових слів 0, будь-яка можлива комбінація  $n$ -бітових слів існує десь у послідовності.

### 3.2.5 Допоміжні модулі cordic та sin\_cos

Взаємопов'язані модулі, що реалізують алгоритм Волдера, що являється ядром для розрахунку синуса та косинуса методом додавання та зсуву. Спрощена блок-схема алгоритму Волдера приведена на рисунку 3.6

					ІА/Ц.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

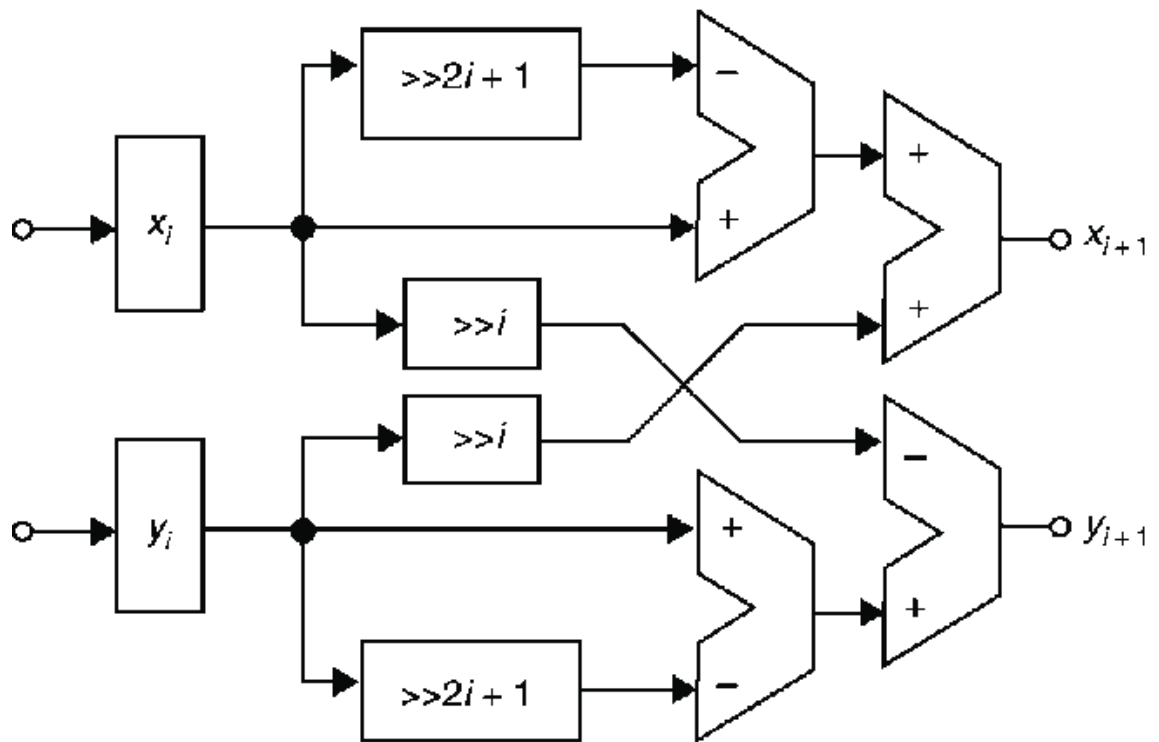


Рис. 3.6

Модуль `sin_cos` перетворює вхідні данні у придатну до обробки форму, після чого викликає модуль `cordic`. Даний модуль проводить розрахунок значення відповідної функції в режимі обертання за наступним принципом:

У першій ітерації вектор обертається на  $45^\circ$  проти годинникової стрілки, щоб отримати вектор  $v_1$ . Послідовні ітерації обертають вектор в ту чи іншу сторону кроками, що зменшують розмір, до досягнення бажаного кута. Крок  $i$  для  $i=0,1,2,\dots$

Більш формально кожна ітерація обчислює обертання, яке виконується шляхом множення вектора  $v_i$  з матрицею обертання  $R_i$ :

$$v_i = R_i v_{i-1} \quad (3.5)$$

Матриця обертання задається числом

$$R_i = \begin{bmatrix} \cos(\gamma_i) & -\sin(\gamma_i) \\ \sin(\gamma_i) & \cos(\gamma_i) \end{bmatrix} \quad (3.6)$$

За допомогою тригонометричних тотожностей данна матриця перетворюється на матрицю обертання.

$$\cos(\gamma_i) = \frac{1}{\sqrt{1 + \tan^2(\gamma_i)}} \quad (3.7)$$

$$\sin(\gamma_i) = \frac{\tan(\gamma_i)}{\sqrt{1 + \tan^2(\gamma_i)}} \quad (3.8)$$

Після достатньої кількості ітерацій кут вектора буде близьким до потрібного кута бета . Для більшості звичайних цілей достатньо 40 ітерацій (  $n = 40$ ) для отримання правильного результату до десятого десяткового знаку.  $\Delta f$

Залишилося єдине завдання - визначити, чи повинен обертання бути за годинниковою або проти годинникової стрілки при кожній ітерації (вибір значення  $\sigma$  ). Це робиться шляхом відстеження того, наскільки кут повернуто при кожній ітерації, і віднімання цього від шуканого кута; то для того, щоб наблизитися до шуканого кута  $\beta$  , якщо  $\beta (n + 1)$  позитивне, обертання - за годинниковою стрілкою, інакше - негативне, а обертання - проти годинникової стрілки.

Після розрахунку за вище описаним алгоритмом данні повертаються в модуль `sin_cos` для подальшої передачі і використання.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

### ВИСНОВКИ ДО РОЗДІЛУ 3

1. Було розроблено програмний компонент системи цифрового радіозв'язку.
2. Для реалізації програмної системи її було поділено на функціональні модулі, кожен з яких реалізовує свою частину необхідних перетворень.
3. Було створено декілька допоміжних модулів, що забезпечують роботу системи та виконання специфічних операцій.

					ІАЛЦ.467100.003 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 4

### СИМУЛЯЦІЯ РОБОТИ СИСТЕМИ

#### 4.1 Результати симуляції виконання програми та аналіз її результатів

Було проведено симуляцію роботи системи за допомогою засобів середовища Active HDL. Результатом симуляції являються графіки, на яких наведено значення сигналів, знятих на різних етапах обробки повідомлення.

Перш за все, відбувається квадратурна модуляція (рис 4.1), що виражено видозміною графіку сигналу. Для більшої наочності, можна порівняти приведений тут результат з схематичним зображенням, приведеним в розділі 1.



Рис. 4.1: результат роботи модулятора

Після квадратурної модуляції відбувається формування скрамблюючих кодів, що відображено на рис. 3.2.

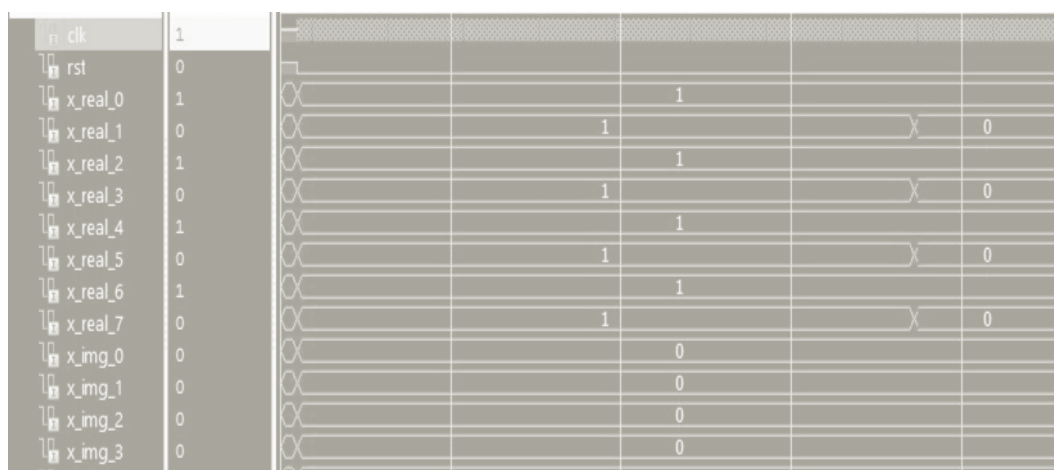


Рис. 3.2

На стороні приймача, в той же час, відбуваються зворотні процеси. Модуль scrambler переключається у зворотній режим роботи, і починає відновлювати зашифровані дані до початкового вигляду, з метою їх подальшої демодуляції. Результат його роботи продемонстровано на рис. 3.3.



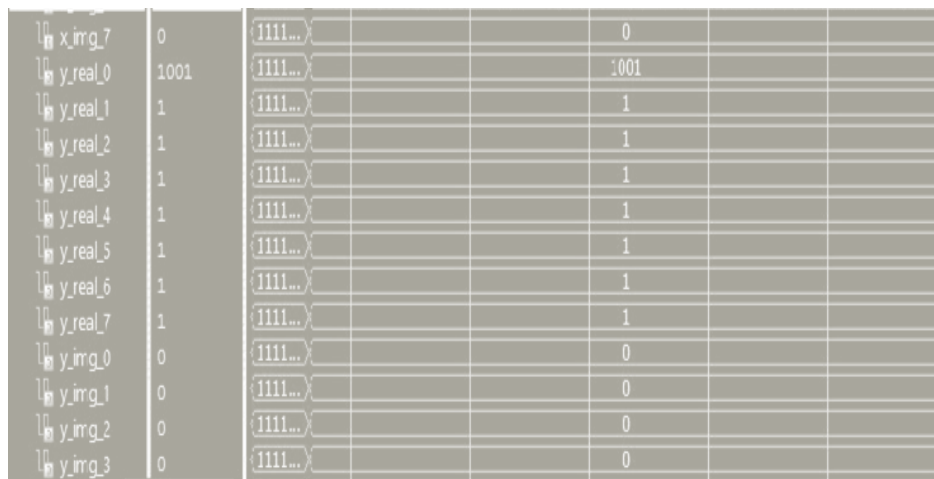


Рис. 3.3

Після дескрамблювання інформації відбувається її демодуляція, що реалізовано за допомогою відповідних модулів. Як видно з рис. 3.4, інформація повертається до початкового вигляду, що свідчить про коректну обробку інформації системою.



Рис. 3.4

Результат симуляції в цілому співпадає з очікуваним результатом виконання протоколу WCDMA. Данні графіку свідчать про перетворення цифрового представлення інформації, поданної з генератора, і співпадають з результатами роботи відповідних модулів програми.

Більш розгорнуте тестування можна буде провести при наявності фізичного компоненту системи.

## 4.2 Інструкція користувача

Для запуску симуляції системи необхідно відкрити файл проекту за допомогою програми Active HDL. Після цього з'явиться інтерфейс проекту (рис. 4.2).

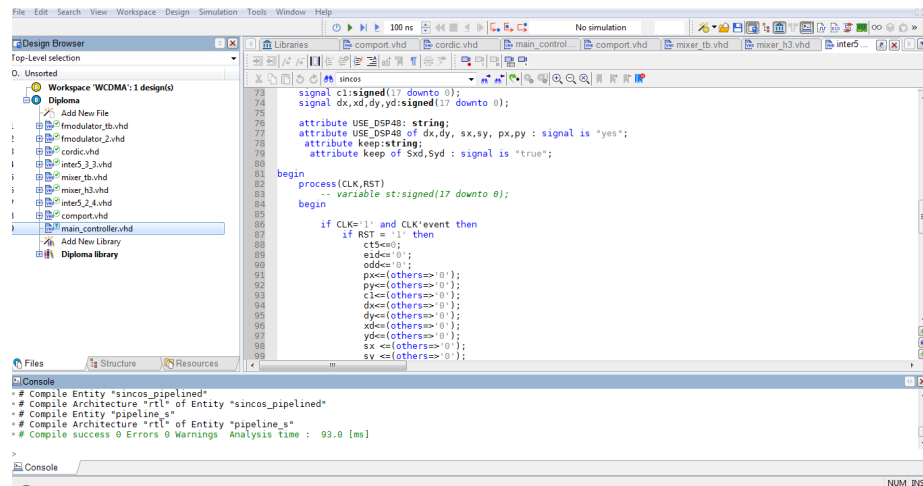


Рис 4.2: проект, відкритий за допомогою Active HDL

Для запуску симуляції необхідно обрати модуль `main_controller.vhd` в якості top-level design unit у відповідному вікні програми (рис 3)

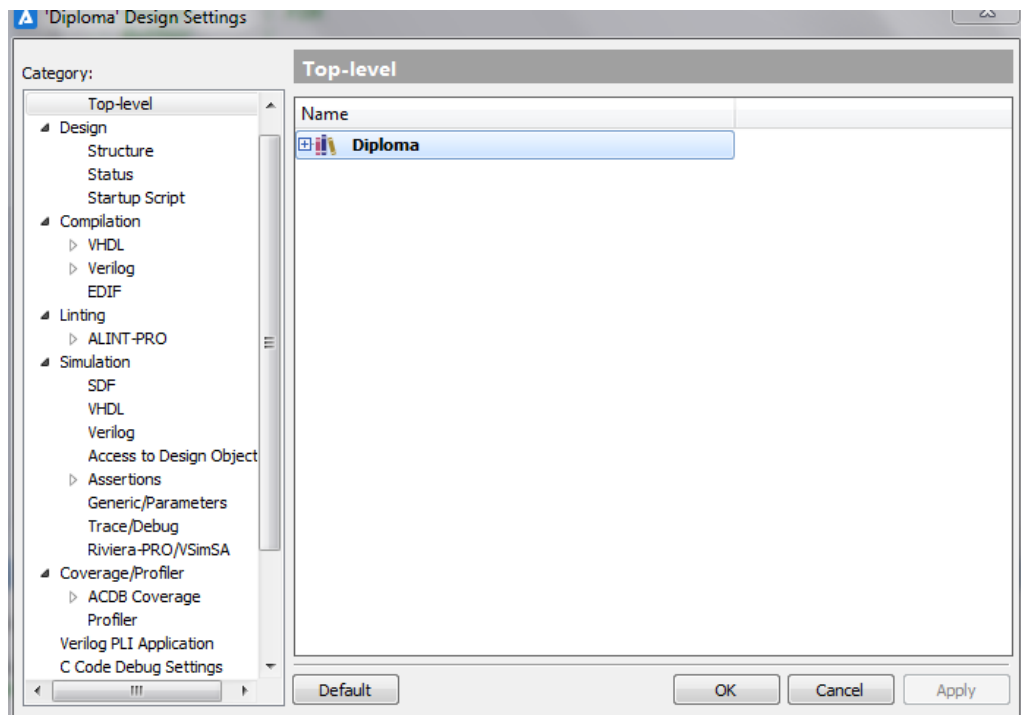


Рис. 4.3

Після цього можна запустити симуляцію роботи програми за допомогою кнопки Run інтерфейсу середовища Active HDL (рис. 4.4), або сполучення клавіш `Alt+F5`.



Рис 4.4

Зм.	Арк.	№ докум.	Підпис	Дата

Результатом симуляції є графік, на якому приведено зняті з різних портів значення сигналів. Якщо графік не відкрився автоматично, треба в ручну відкрити відповідний файл з каталогу проекту (рис. 4.5) Ідентичні вхідний та вихідний сигнал свідчать про коректну роботу програми.

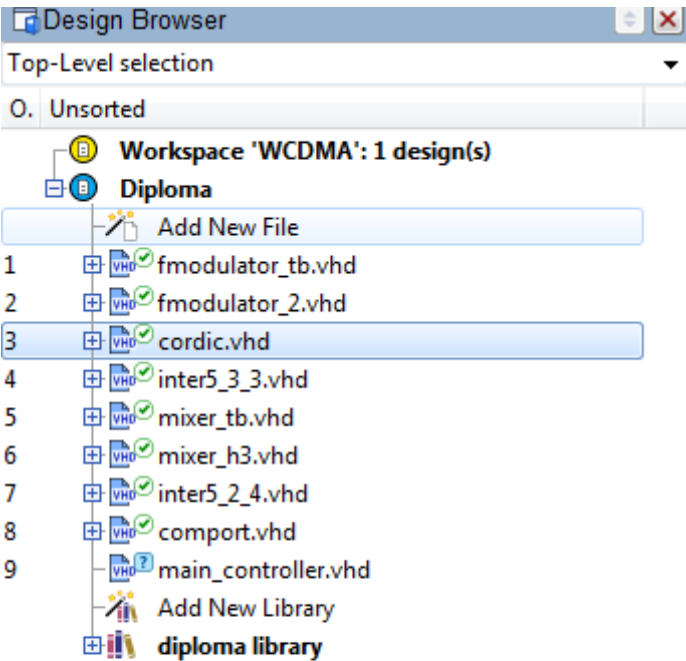


Рис. 4.5

#### ВИСНОВКИ ДО РОЗДІЛУ 4

1. Було проведено симуляцію роботи системи засобами симуляційного середовища Active HDL.
2. Результатти симуляції співпадають з ймовірними результатами роботи системи цифрового радіозв'язку, що реалізує стандарт WCDMA. Це виражено у вигляді поступового перетворення даних шляхом модуляції, котування, та підсилення. Графіки, побудовані на базі значень сигналів на портах приймача, свідчать про коректні зворотні перетворення сигналу.

					ІАЛЦ.467100.003 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

1. У данній роботі було розглянуто особливості цифрових систем радіозв'язку, їх особливості, та способи реалізації. Також було проведено огляд різноманітних тематичних стандартів, аналіз їх особливостей. Було обгрунтовано необхідність створення програмно-визначеної системи зв'язку, що реалізує стандарт WCDMA.
2. Було розроблено програмний компонент системи за допомогою засобів мови програмування VHDL. Також розроблена програма реалізує емуляцію роботи апаратної частини засобами Active HDL.
3. Було проведено симуляцію роботи системи, за результатами якої було визначено, що система загалом працює коректно, і реалізує передачу сигналу згідно стандарту WCDMA.

					ІАЛЦ.467100.003 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ПОСИЛАНЬ

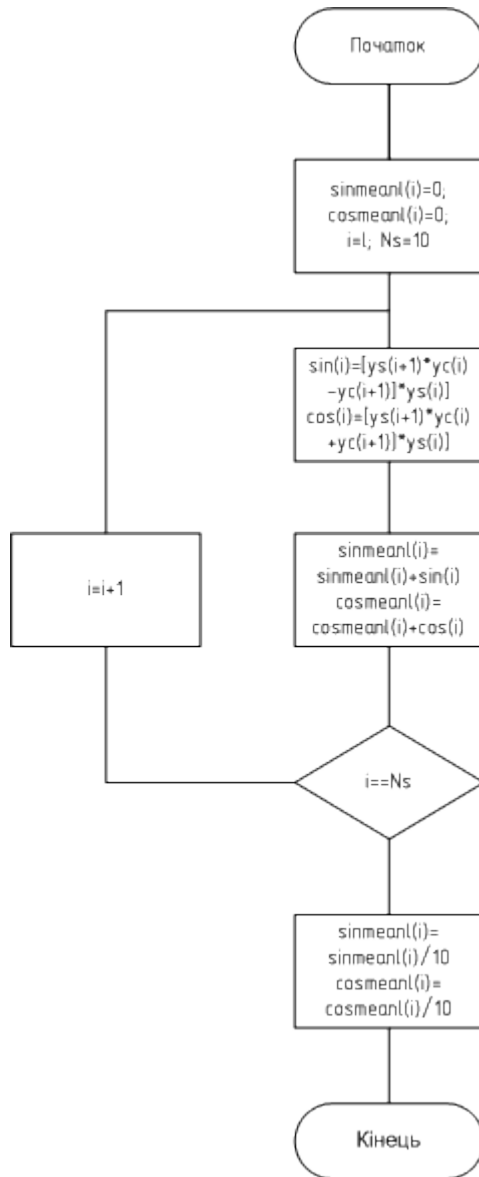
1. Grebennikov A. Rf And Microwave Transmitter Design / Andrei Grebennikov. – New Jersey: John Wiley & Sons, 2011. – 839 p.
2. Yang M. S. Modern Digital Radio Communication Signals and Systems / Michael Sung-Moon Yang. – Irvine: Springer International Publishing AG, 2019. – 677 p.
3. Software-defined Radios: Architecture, State-of-the-art, and Challenges / Rami Akeela, Behnam Dezfouli – Santa clara: Santa clara university, 2018 – 23 p.
4. Freeman R. L. Radio System Design for Telecommunications / Roger L. Freeman. – New Jersey: John Wiley & Sons, 2007. – 911 p.
5. Software-Defined Radio for Engineers / T. F.Collins, R. Getz, D. Pu, A. M. Wyglinski., 2018. – 375 p.
6. LimeSDR USB User Guide [Електронний ресурс] // Lime microsystems. – Режим доступу до ресурсу: [https://wiki.myriadrfr.org/LimeSDR-USB\\_User\\_Guide](https://wiki.myriadrfr.org/LimeSDR-USB_User_Guide)
7. Kalivas G. Digital radio system design / Grigorios Kalivas. – Singapore: JohnWiley & Sons Ltd, 2009. – 474 p.
8. Digital Radio Guide / A. Punchihewa, A. Zink, L. Sabel et al., 2014. – 145 p.
9. Недяев Л. М. 6. Мобильная связь 3-го поколения / Леонид Михайлович Недяев. – Москва: ООО "Мобильные коммуникации", 2000. – 208 с.
10. Richardson A. WCDMA Design Handbook / Andrew Richardson. – Cambridge: Cambridge university press, 2005. – 593 p.
11. LMS7002M Datasheet [Електронний ресурс] // Lime microsystems. – 2014. – Режим доступу до ресурсу: [https://github.com/myriadrfr/LMS7002M-docs/blob/master/LMS7002M\\_Data\\_Sheet\\_v3.2r00.pdf](https://github.com/myriadrfr/LMS7002M-docs/blob/master/LMS7002M_Data_Sheet_v3.2r00.pdf).
12. LimeSDR-USB hardware description [Електронний ресурс] // Lime microsystems. – Режим доступу до ресурсу: [https://wiki.myriadrfr.org/LimeSDR-USB\\_hardware\\_description45u56rtg](https://wiki.myriadrfr.org/LimeSDR-USB_hardware_description45u56rtg)

13. LimeSDR-USB Board Programming [Електронний ресурс] // Lime microsystems. – Режим доступу до ресурсу: [https://wiki.myriadrf.org/LimeSDR-USB\\_Board\\_Programming](https://wiki.myriadrf.org/LimeSDR-USB_Board_Programming)
14. LMS7002M Programming and Calibration Guide [Електронний ресурс] // Lime microsystems. – Режим доступу до ресурсу: [https://github.com/myriadrf/LMS7002M-docs/blob/master/LMS7002M\\_Programming\\_and\\_Calibration\\_Guide\\_v31r05.pdf](https://github.com/myriadrf/LMS7002M-docs/blob/master/LMS7002M_Programming_and_Calibration_Guide_v31r05.pdf)
15. Stanev V. Abtastratenumsetzung nach dem Direct-Down-Conversion Prinzip als FPGA-IP-Core für nachfolgende Auswertung durch Matlab / Valentin Stanev – Hamburg: Hamburg University of Applied Sciences, 2011 – 102 p.
16. Digital Radio Broadcasting Using LimeSDR and ODR Tools [Електронний ресурс] / Godfrey L. – Режим доступу до ресурсу: <https://medium.com/@sundayglee/digital-radio-transmission-using-limesdr-and-odr-tools-part-2-8f445fedde75>

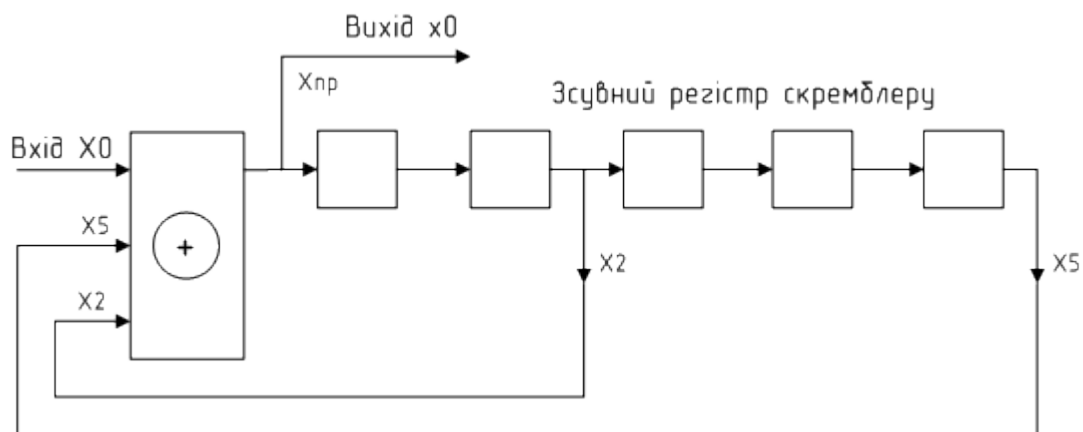


					ІАЛЦ.467800.005 ДА										
					Цифрова система радіозв'язку Структура системи. Схема структурна.										
Зм.	Аркуш	№ Докум.	Підпис	Дата											
Розробив		Кородейніков Д. Ю.													
Перевірив		Сергієнко А. С.													
Т. Контр.															
					Дипломний проект					Аркуш 1		Аркушів 1			
Н. контр.		Сімоненко В.П.								НТУУ “КПІ”, ФІОТ, Ю-62					
Затвердив															





					ІАЛЦ.467800.005 ДБ									
					Цифрова система радіозв'язку Алгоритм CORDIC. Схема алгоритму.									
Зм.	Аркуш	№ Докум.	Підпис	Дата										
Розробив		Кородейніков Д. Ю.												
Перевірів		Сергієнко А. С.												
Т. Контр.														
					Дипломний проект	Аркуш 1		Аркушів 1						
Н. контр.		Сімоненко В.П.				НТУУ “КПІ”, ФІОТ, Ю-62								
Затвердив														



					ІА/Ц.467800.005 ДВ					
					Цифрова система радіозв'язку Формувач скрамблюючого коду. Схема функціональна.					
Зм.	Аркуш	№ Докум.	Підпис	Дата						
Розробив		Кородейніков Д. Ю.								
Перевірив		Сергієнко А. С.								
Т. Контр.					Аркуш 1      Аркушів 1					
Н. контр.		Сімоненко В.П.			Дипломний проект					
Затвердив										
					НТУУ "КПІ", ФІОТ, ІО-62					

## ДОДАТОК Г

### КОД ПРОГРАМНОЇ ЧАСТИНИ

```
library ieee;
use ieee.NUMERIC_STD.all;
use ieee.std_logic_1164.all;
use ieee.math_real.all;

-- Add your library and packages declaration here ...

entity fmodulator_tb is
end fmodulator_tb;

architecture TB_ARCHITECTURE of fmodulator_tb is
-- Component declaration of the tested unit
component fmodulator
port(
    CLK : in STD_LOGIC;
    RST : in STD_LOGIC;
    EI : in STD_LOGIC;
    D_I : in STD_LOGIC_VECTOR(23 downto 0);
    AMPLIF : in STD_LOGIC_VECTOR(15 downto 0);
    MAGN : in STD_LOGIC_VECTOR(15 downto 0);
    EO : out STD_LOGIC;
    E192 : out STD_LOGIC;
    RE_O : out STD_LOGIC_VECTOR(17 downto 0);
    IM_O : out STD_LOGIC_VECTOR(17 downto 0) );
end component;

-- Stimulus signals - signals mapped to the input and inout ports of tested entity
signal CLK : STD_LOGIC:= '0';
signal RST : STD_LOGIC;
signal EI : STD_LOGIC;
signal D_I : STD_LOGIC_VECTOR(23 downto 0);
signal AMPLIF : STD_LOGIC_VECTOR(15 downto 0);
signal MAGN : STD_LOGIC_VECTOR(15 downto 0);
-- Observed signals - signals mapped to the output ports of tested entity
signal EO : STD_LOGIC;
signal E192 : STD_LOGIC;
signal RE_O : STD_LOGIC_VECTOR(17 downto 0);
signal IM_O : STD_LOGIC_VECTOR(17 downto 0);
signal K,db:real;

begin

    db<=13.084;

    k<= exp(db*log(10.0)/20.0);--/4.5102;

    clk<= not clk after 10 ns;
    rst<= '1', '0' after 44 ns;

    process begin
        ei<= '1' ;
```

```

wait for 20 ns;
ei<='0' ;
wait for 5.2 us;

```

```

end process;

```

```

d_i<=x"711333", x"c11333" after 140 us;
MAGN<=x"2fff", x"4000" after 240 us;
AMPLIF<=x"8000", x"4000" after 340 us;

```

```

UUT : fmodulator
    port map (
        CLK => CLK,
        RST => RST,
        EI => EI,
        D_I => D_I,
        AMPLIF => AMPLIF,
        MAGN => MAGN,
        EO => EO,
        E192 => E192,
        RE_O => RE_O,
        IM_O => IM_O
    );

```

```

end TB_ARCHITECTURE;

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.numeric_STD.all;
entity FMODULATOR is
    port(
        CLK : in STD_LOGIC;
        RST : in STD_LOGIC;
        EI : in STD_LOGIC;
        D_I : in STD_LOGIC_VECTOR(23 downto 0);
        AMPLIF : in STD_LOGIC_VECTOR(15 downto 0);-- input amplification
        MAGN : in STD_LOGIC_VECTOR(15 downto 0); -- output amplification
        EO : out STD_LOGIC;
        E192 : out STD_LOGIC;
        RE_O : out STD_LOGIC_VECTOR(17 downto 0);
        IM_O : out STD_LOGIC_VECTOR(17 downto 0)
    );
end FMODULATOR;

```

```

architecture synt of FMODULATOR is

```

```

    --ROM of 1/2 period of a sine wave

```

```

    type arr1024 is array (0 to 1023) of STD_LOGIC_VECTOR(15 downto 0);

```

```
signal sin_arr:arr1024:=(  
X"0000",X"00C9",X"0192",X"025B",  
X"0324",X"03ED",X"04B6",X"057F",  
X"0648",X"0711",X"07DA",X"08A3",  
X"096C",X"0A35",X"0AFD",X"0BC6",  
X"0C8F",X"0D58",X"0E21",X"0EEA",  
X"0FB2",X"107B",X"1143",X"120C",  
X"12D5",X"139D",X"1466",X"152E",  
X"15F6",X"16BF",X"1787",X"184F",  
X"1917",X"19DF",X"1AA7",X"1B6F",  
X"1C37",X"1CFF",X"1DC7",X"1E8E",  
X"1F56",X"201D",X"20E5",X"21AC",  
X"2273",X"233B",X"2402",X"24C9",  
X"2590",X"2656",X"271D",X"27E4",  
X"28AA",X"2971",X"2A37",X"2AFD",  
X"2BC4",X"2C8A",X"2D50",X"2E15",  
X"2EDB",X"2FA1",X"3066",X"312C",  
X"31F1",X"32B6",X"337B",X"3440",  
X"3505",X"35C9",X"368E",X"3752",  
X"3816",X"38DB",X"399F",X"3A62",  
X"3B26",X"3BEA",X"3CAD",X"3D70",  
X"3E33",X"3EF6",X"3FB9",X"407C",  
X"413E",X"4201",X"42C3",X"4385",  
X"4447",X"4508",X"45CA",X"468B",  
X"474D",X"480E",X"48CE",X"498F",  
X"4A50",X"4B10",X"4BD0",X"4C90",  
X"4D50",X"4E0F",X"4ECF",X"4F8E",  
X"504D",X"510C",X"51CA",X"5289",  
X"5347",X"5405",X"54C3",X"5580",  
X"563E",X"56FB",X"57B8",X"5875",  
X"5931",X"59EE",X"5AAA",X"5B66",  
X"5C22",X"5CDD",X"5D98",X"5E53",  
X"5F0E",X"5FC9",X"6083",X"613D",  
X"61F7",X"62B1",X"636A",X"6423",  
X"64DC",X"6595",X"664D",X"6705",  
X"67BD",X"6875",X"692C",X"69E4",  
X"6A9B",X"6B51",X"6C08",X"6CBE",  
X"6D74",X"6E29",X"6EDF",X"6F94",  
X"7049",X"70FD",X"71B1",X"7265",  
X"7319",X"73CD",X"7480",X"7533",  
X"75E5",X"7698",X"774A",X"77FB",  
X"78AD",X"795E",X"7A0F",X"7AC0",  
X"7B70",X"7C20",X"7CD0",X"7D7F",  
X"7E2E",X"7EDD",X"7F8B",X"8039",  
X"80E7",X"8195",X"8242",X"82EF",  
X"839C",X"8448",X"84F4",X"85A0",  
X"864B",X"86F6",X"87A1",X"884B",  
X"88F5",X"899F",X"8A48",X"8AF1",  
X"8B9A",X"8C42",X"8CEA",X"8D92",  
X"8E39",X"8EE0",X"8F87",X"902D",  
X"90D3",X"9179",X"921E",X"92C3",  
X"9368",X"940C",X"94B0",X"9553",  
X"95F6",X"9699",X"973B",X"97DE",  
X"987F",X"9921",X"99C1",X"9A62",  
X"9B02",X"9BA2",X"9C41",X"9CE1",
```

X"9D7F",X"9E1E",X"9EBC",X"9F59",  
X"9FF6",X"A093",X"A12F",X"A1CB",  
X"A267",X"A302",X"A39D",X"A438",  
X"A4D2",X"A56B",X"A604",X"A69D",  
X"A736",X"A7CE",X"A865",X"A8FD",  
X"A994",X"AA2A",X"AAC0",X"AB56",  
X"ABEB",X"AC7F",X"AD14",X"ADA8",  
X"AE3B",X"AECE",X"AF61",X"AFF3",  
X"B085",X"B117",X"B1A8",X"B238",  
X"B2C8",X"B358",X"B3E7",X"B476",  
X"B504",X"B592",X"B620",X"B6AD",  
X"B73A",X"B7C6",X"B852",X"B8DD",  
X"B968",X"B9F2",X"BA7C",X"BB06",  
X"BB8F",X"BC17",X"BC9F",X"BD27",  
X"BDAE",X"BE35",X"BEBC",X"BF41",  
X"BFC7",X"C04C",X"C0D0",X"C154",  
X"C1D8",X"C25B",X"C2DE",X"C360",  
X"C3E1",X"C463",X"C4E3",X"C564",  
X"C5E3",X"C663",X"C6E2",X"C760",  
X"C7DE",X"C85B",X"C8D8",X"C955",  
X"C9D0",X"CA4C",X"CAC7",X"CB41",  
X"CBBB",X"CC35",X"CCAE",X"CD26",  
X"CD9E",X"CE16",X"CE8D",X"CF04",  
X"CF7A",X"CFEF",X"D064",X"D0D9",  
X"D14D",X"D1C0",X"D233",X"D2A6",  
X"D318",X"D389",X"D3FA",X"D46B",  
X"D4DB",X"D54A",X"D5B9",X"D627",  
X"D695",X"D703",X"D770",X"D7DC",  
X"D848",X"D8B3",X"D91E",X"D988",  
X"D9F2",X"DA5B",X"DAC4",X"DB2C",  
X"DB94",X"DBFB",X"DC61",X"DCC7",  
X"DD2D",X"DD92",X"DDF6",X"DE5A",  
X"DEBD",X"DF20",X"DF83",X"DFE4",  
X"E046",X"E0A6",X"E106",X"E166",  
X"E1C5",X"E224",X"E281",X"E2DF",  
X"E33C",X"E398",X"E3F4",X"E44F",  
X"E4AA",X"E504",X"E55D",X"E5B6",  
X"E60F",X"E667",X"E6BE",X"E715",  
X"E76B",X"E7C1",X"E816",X"E86B",  
X"E8BF",X"E912",X"E965",X"E9B7",  
X"EA09",X"EA5A",X"EAAB",X"EAFB",  
X"EB4A",X"EB99",X"EBE8",X"EC36",  
X"EC83",X"ECCF",X"ED1C",X"ED67",  
X"EDB2",X"EDFC",X"EE46",X"EE8F",  
X"EED8",X"EF20",X"EF68",X"EFAF",  
X"EFF5",X"F03B",X"F080",X"F0C4",  
X"F108",X"F14C",X"F18F",X"F1D1",  
X"F213",X"F254",X"F294",X"F2D4",  
X"F314",X"F352",X"F391",X"F3CE",  
X"F40B",X"F448",X"F484",X"F4BF",  
X"F4F9",X"F534",X"F56D",X"F5A6",  
X"F5DE",X"F616",X"F64D",X"F684",  
X"F6B9",X"F6EF",X"F723",X"F758",  
X"F78B",X"F7BE",X"F7F0",X"F822",  
X"F853",X"F884",X"F8B4",X"F8E3",

X"F912",X"F940",X"F96E",X"F99B",  
X"F9C7",X"F9F3",X"FA1E",X"FA48",  
X"FA72",X"FA9C",X"FAC4",X"FAED",  
X"FB14",X"FB3B",X"FB61",X"FB87",  
X"FBAC",X"FBD1",X"FBF5",X"FC18",  
X"FC3B",X"FC5D",X"FC7E",X"FC9F",  
X"FCBF",X"FCDF",X"FCFE",X"FD1C",  
X"FD3A",X"FD57",X"FD74",X"FD90",  
X"FDAB",X"FDC6",X"FDE0",X"FDFA",  
X"FE13",X"FE2B",X"FE43",X"FE5A",  
X"FE70",X"FE86",X"FE9B",X"FEB0",  
X"FEC4",X"FED7",X"FEEA",X"FEFC",  
X"FF0E",X"FF1F",X"FF2F",X"FF3F",  
X"FF4E",X"FF5C",X"FF6A",X"FF77",  
X"FF84",X"FF90",X"FF9B",X"FFA6",  
X"FFB0",X"FFBA",X"FFC3",X"FFCB",  
X"FFD3",X"FFDA",X"FFE1",X"FFE6",  
X"FFEC",X"FFF0",X"FFF4",X"FFF8",  
X"FFFA",X"FFFD",X"FFFE",X"FFFF",  
X"FFFF",X"FFFF",X"FFFE",X"FFFD",  
X"FFFA",X"FFF8",X"FFF4",X"FFF0",  
X"FFEC",X"FFE6",X"FFE1",X"FFDA",  
X"FFD3",X"FFCB",X"FFC3",X"FFBA",  
X"FFB0",X"FFA6",X"FF9B",X"FF90",  
X"FF84",X"FF77",X"FF6A",X"FF5C",  
X"FF4E",X"FF3F",X"FF2F",X"FF1F",  
X"FF0E",X"FEFC",X"FEEA",X"FED7",  
X"FEC4",X"FEB0",X"FE9B",X"FE86",  
X"FE70",X"FE5A",X"FE43",X"FE2B",  
X"FE13",X"FDFA",X"FDE0",X"FDC6",  
X"FDAB",X"FD90",X"FD74",X"FD57",  
X"FD3A",X"FD1C",X"FCFE",X"FCDF",  
X"FCBF",X"FC9F",X"FC7E",X"FC5D",  
X"FC3B",X"FC18",X"FBF5",X"FBD1",  
X"FBAC",X"FB87",X"FB61",X"FB3B",  
X"FB14",X"FAED",X"FAC4",X"FA9C",  
X"FA72",X"FA48",X"FA1E",X"F9F3",  
X"F9C7",X"F99B",X"F96E",X"F940",  
X"F912",X"F8E3",X"F8B4",X"F884",  
X"F853",X"F822",X"F7F0",X"F7BE",  
X"F78B",X"F758",X"F723",X"F6EF",  
X"F6B9",X"F684",X"F64D",X"F616",  
X"F5DE",X"F5A6",X"F56D",X"F534",  
X"F4F9",X"F4BF",X"F484",X"F448",  
X"F40B",X"F3CE",X"F391",X"F352",  
X"F314",X"F2D4",X"F294",X"F254",  
X"F213",X"F1D1",X"F18F",X"F14C",  
X"F108",X"F0C4",X"F080",X"F03B",  
X"EFF5",X"EF AF",X"EF68",X"EF20",  
X"EED8",X"EE8F",X"EE46",X"EDFC",  
X"EDB2",X"ED67",X"ED1C",X"EC CF",  
X"EC83",X"EC36",X"EBE8",X"EB99",  
X"EB4A",X"EAFB",X"EAAB",X"EA5A",  
X"EA09",X"E9B7",X"E965",X"E912",  
X"E8BF",X"E86B",X"E816",X"E7C1",

X"E76B",X"E715",X"E6BE",X"E667",  
X"E60F",X"E5B6",X"E55D",X"E504",  
X"E4AA",X"E44F",X"E3F4",X"E398",  
X"E33C",X"E2DF",X"E281",X"E224",  
X"E1C5",X"E166",X"E106",X"E0A6",  
X"E046",X"DFE4",X"DF83",X"DF20",  
X"DEBD",X"DE5A",X"DDF6",X"DD92",  
X"DD2D",X"DCC7",X"DC61",X"DBFB",  
X"DB94",X"DB2C",X"DAC4",X"DA5B",  
X"D9F2",X"D988",X"D91E",X"D8B3",  
X"D848",X"D7DC",X"D770",X"D703",  
X"D695",X"D627",X"D5B9",X"D54A",  
X"D4DB",X"D46B",X"D3FA",X"D389",  
X"D318",X"D2A6",X"D233",X"D1C0",  
X"D14D",X"D0D9",X"D064",X"CFEF",  
X"CF7A",X"CF04",X"CE8D",X"CE16",  
X"CD9E",X"CD26",X"CCAE",X"CC35",  
X"CBBB",X"CB41",X"CAC7",X"CA4C",  
X"C9D0",X"C955",X"C8D8",X"C85B",  
X"C7DE",X"C760",X"C6E2",X"C663",  
X"C5E3",X"C564",X"C4E3",X"C463",  
X"C3E1",X"C360",X"C2DE",X"C25B",  
X"C1D8",X"C154",X"C0D0",X"C04C",  
X"BFC7",X"BF41",X"BEBC",X"BE35",  
X"BDAE",X"BD27",X"BC9F",X"BC17",  
X"BB8F",X"BB06",X"BA7C",X"B9F2",  
X"B968",X"B8DD",X"B852",X"B7C6",  
X"B73A",X"B6AD",X"B620",X"B592",  
X"B504",X"B476",X"B3E7",X"B358",  
X"B2C8",X"B238",X"B1A8",X"B117",  
X"B085",X"AFF3",X"AF61",X"AECE",  
X"AE3B",X"ADA8",X"AD14",X"AC7F",  
X"ABEB",X"AB56",X"AAC0",X"AA2A",  
X"A994",X"A8FD",X"A865",X"A7CE",  
X"A736",X"A69D",X"A604",X"A56B",  
X"A4D2",X"A438",X"A39D",X"A302",  
X"A267",X"A1CB",X"A12F",X"A093",  
X"9FF6",X"9F59",X"9EBC",X"9E1E",  
X"9D7F",X"9CE1",X"9C41",X"9BA2",  
X"9B02",X"9A62",X"99C1",X"9921",  
X"987F",X"97DE",X"973B",X"9699",  
X"95F6",X"9553",X"94B0",X"940C",  
X"9368",X"92C3",X"921E",X"9179",  
X"90D3",X"902D",X"8F87",X"8EE0",  
X"8E39",X"8D92",X"8CEA",X"8C42",  
X"8B9A",X"8AF1",X"8A48",X"899F",  
X"88F5",X"884B",X"87A1",X"86F6",  
X"864B",X"85A0",X"84F4",X"8448",  
X"839C",X"82EF",X"8242",X"8195",  
X"80E7",X"8039",X"7F8B",X"7EDD",  
X"7E2E",X"7D7F",X"7CD0",X"7C20",  
X"7B70",X"7AC0",X"7A0F",X"795E",  
X"78AD",X"77FB",X"774A",X"7698",  
X"75E5",X"7533",X"7480",X"73CD",  
X"7319",X"7265",X"71B1",X"70FD",



```

X"7049",X"6F94",X"6EDF",X"6E29",
X"6D74",X"6CBE",X"6C08",X"6B51",
X"6A9B",X"69E4",X"692C",X"6875",
X"67BD",X"6705",X"664D",X"6595",
X"64DC",X"6423",X"636A",X"62B1",
X"61F7",X"613D",X"6083",X"5FC9",
X"5F0E",X"5E53",X"5D98",X"5CDD",
X"5C22",X"5B66",X"5AAA",X"59EE",
X"5931",X"5875",X"57B8",X"56FB",
X"563E",X"5580",X"54C3",X"5405",
X"5347",X"5289",X"51CA",X"510C",
X"504D",X"4F8E",X"4ECF",X"4E0F",
X"4D50",X"4C90",X"4BD0",X"4B10",
X"4A50",X"498F",X"48CE",X"480E",
X"474D",X"468B",X"45CA",X"4508",
X"4447",X"4385",X"42C3",X"4201",
X"413E",X"407C",X"3FB9",X"3EF6",
X"3E33",X"3D70",X"3CAD",X"3BEA",
X"3B26",X"3A62",X"399F",X"38DB",
X"3816",X"3752",X"368E",X"35C9",
X"3505",X"3440",X"337B",X"32B6",
X"31F1",X"312C",X"3066",X"2FA1",
X"2EDB",X"2E15",X"2D50",X"2C8A",
X"2BC4",X"2AFD",X"2A37",X"2971",
X"28AA",X"27E4",X"271D",X"2656",
X"2590",X"24C9",X"2402",X"233B",
X"2273",X"21AC",X"20E5",X"201D",
X"1F56",X"1E8E",X"1DC7",X"1CFF",
X"1C37",X"1B6F",X"1AA7",X"19DF",
X"1917",X"184F",X"1787",X"16BF",
X"15F6",X"152E",X"1466",X"139D",
X"12D5",X"120C",X"1143",X"107B",
X"0FB2",X"0EEA",X"0E21",X"0D58",
X"0C8F",X"0BC6",X"0AFD",X"0A35",
X"096C",X"08A3",X"07DA",X"0711",
X"0648",X"057F",X"04B6",X"03ED",
X"0324",X"025B",X"0192",X"00C9"
);

```

```

signal phase,freq_code: signed(23 downto 0);
signal phre,phim,phred,phimd:std_logic_vector(10 downto 0);
signal sgnre,sgnim,sgnred,sgnimd,sgnred2,sgnimd2,e192i, even:std_logic;
signal sine,cosine,sind,cosd: signed(16 downto 0);
signal ct100: natural range 0 to 99;
signal ct5: natural range 0 to 4;
signal p:signed(40 downto 0); -- 24x17=41
signal D_r: STD_LOGIC_VECTOR(23 downto 0);
signal lr: Signed(23 downto 0);
signal rr,amp,mag : Signed(16 downto 0);

```

```

attribute maxdelay: string;
attribute maxdelay of p,lr,rr : signal is "24 mhz";
attribute RAM_STYLE : string;
-- attribute RAM_STYLE of sin_arr: signal is "DISTRIBUTED";

```

begin

-- scaler

SCALER: process(CLK,RST)

variable l:signed(23 downto 0):=x"000000";  
variable r:signed(16 downto 0):='0'&x"0000";  
variable pi:signed(40 downto 0);

begin

if CLK='1' and CLK'event then

if RST='1' then

p <= (others =>'0');  
d\_r <= (others =>'0');  
RE\_O <=(others =>'0');  
IM\_O <=(others =>'0');  
freq\_code<=(others =>'0');  
lr<= (others =>'0');  
rr<= (others =>'0');  
amp<= (others =>'0');  
mag<= (others =>'0');

-- even<='0';  
EO<='0';

else

amp<=signed('0'&AMPLIF);  
mag<=signed('0'&MAGN);  
even<=not even;

-- if even = '1' then

if EI='1' then

d\_r<=D\_I;

end if;

EO<='0';

case ct100 is

when 2 => --p<= signed(D\_I)\*signed('0'&AMPLIF);  
lr<=signed(D\_r);

rr<= amp;

when 3 => --p<= \*signed('0'&MAGN);

when 4 => --p<= (sine&"000"&x"0")\*signed('0'&MAGN);

lr<=(cosine&"000"&x"0");

freq\_code<= P(39)&P(39)&P(39 downto 18);

rr<=mag;

when 5 =>

-- RE\_O<= std\_logic\_vector(p(38 downto 21));

lr<=(sine&"000"&x"0");

rr<=mag;

when 6 =>

RE\_O<= std\_logic\_vector(P(39 downto 22));

when 7 =>

```

IM_O<= std_logic_vector(P(39 downto 22));

when 8=>
    EO<='1';
when others=> null;
end case;
p<= lr * rr;

--      end if;
    end if;
end if;
end process;

```

-- phase generator clocks 24 MHz/50

```

PH_GEN:process(CLK,RST)
    variable re: signed (11 downto 0);

```

```

begin

```

```

    if CLK='1' and CLK'event then
        if RST='1' then
            phase <=X"000000";
            phre <="010"&x"00";
            phim <="000"&x"00";
            sgnre <='0';
            sgnim <='0';
            ct100 <= 0;
            ct5<= 0;
            e192i<='0';
            --      EO<='0';
        else
            --      EO<='0';
            --      if even ='1' then
                --      if ct100 = 99 then
                    if ct100 = 24 then
                        ct100<=0;
                        --      EO<='1';
                        if ct5 = 4 then
                            e192i<='1';
                            ct5<= 0;
                        else
                            ct5<=ct5+1;
                        end if;
                        phase <=phase + signed(freq_code);
                        phim <= std_logic_vector(phase(22 downto 12));
                        sgnim <= phase(23);
                        re:= phase(23 downto 12) + (x"400");
                        phre <= std_logic_vector(re(10 downto 0));
                        sgnre <= re(11);
                    else

```

```

ct100<=ct100 + 1;
end if;

--      end if;
end if;

end if;
end process;

--Sine-cosine waves
ROM:process(CLK) begin
    if CLK='1' and CLK'event then
        if EI='0' then
            phred<=phre;
            sgnred<=sgnre;
            sgnred2<=sgnred;
            phimd<=phim;
            sgnimd<=sgnim;
            sgnimd2<=sgnimd;

            sind<= signed('0'&sin_arr(to_integer(unsigned(phimd(10 downto 1)))));
            cosd<= signed('0'&sin_arr(to_integer(unsigned(phred(10 downto 1)))));

            if sgnimd2 = '0' then
                sine<= sind;
            else
                sine<=(0 - sind);
            end if;
            if sgnred2 = '0' then
                cosine<= cosd;
            else
                cosine<=(0 - cosd);
            end if;
        end if;
    end if;

end process;

E192<=e192i;

```

end synt

```

procedure rotate(iterations : in integer;
    x, y, z : in signed;
    signal xr, yr, zr : out signed
) is

```

```

    variable xp, yp, xp2, yp2 : signed(x'range);
    variable zp, zp2 : signed(z'range);
    variable z_inc : integer;
begin
    xp := x; yp := y; zp := z;

```

```

    for i in 0 to iterations-1 loop

```

```

z_inc := integer(arctan(2.0**(-i)) * 2.0**z'length / MATH_2_PI);

if zp(zp'high) = '1' then -- negative
    xp2 := xp + yp / 2**i;
    yp2 := yp - xp / 2**i;
    zp2 := zp + z_inc;
else
    xp2 := xp - yp / 2**i;
    yp2 := yp + xp / 2**i;
    zp2 := zp - z_inc;
end if;

xp := xp2; yp := yp2; zp := zp2;
end loop;

xr <= xp; yr <= yp; zr <= zp;
end procedure;

--## Perform CORDIC vector mode operation.
--# This synthesizes to a pure combinational architecture.
procedure vector(iterations : in integer; -- Number of CORDIC iterations
    x, y, z : in signed; -- X,Y point and angle
    signal xr, yr, zr : out signed -- Post-CORDIC result point and angle
) is
    variable xp, yp, xp2, yp2 : signed(x'range);
    variable zp, zp2 : signed(z'range);
    variable z_inc : integer;
begin
    xp := x; yp := y; zp := z;

    for i in 0 to iterations-1 loop

        z_inc := integer(arctan(2.0**(-i)) * 2.0**z'length / MATH_2_PI);

        if yp(yp'high) = '0' then -- positive
            xp2 := xp + yp / 2**i;
            yp2 := yp - xp / 2**i;
            zp2 := zp + z_inc;
        else
            xp2 := xp - yp / 2**i;
            yp2 := yp + xp / 2**i;
            zp2 := zp - z_inc;
        end if;

        xp := xp2; yp := yp2; zp := zp2;
    end loop;

    xr <= xp; yr <= yp; zr <= zp;
end procedure;

function overall_quantization_error(iterations : positive; frac_bits : positive) return real is
    constant k : real := cordic_gain(iterations);
    variable g, p : real;

```

```

    variable approx_error, rounding_error : real;
begin
    approx_error := k - k * cos(arctan(2.0**(-iterations + 1)));

    g := 0.0;
    for j in 1 to iterations - 1 loop
        p := 1.0;
        for i in j to iterations - 1 loop
            p := p * sqrt(1.0 + 2.0**(-2*i));
        end loop;
        g := g + p;
    end loop;

    g := g + 1.0;

    rounding_error := 2.0**(-real(frac_bits) - 0.5)*(g/k + 1.0);

    return approx_error + rounding_error;
end function;

function effective_fractional_bits(iterations, frac_bits : positive) return real is
begin
    return -log2(overall_quantization_error(iterations, frac_bits));
end function;

end package body;

```

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.math_real.all;

```

```

library project;
use project.cordic.all;

```

```

entity cordic_pipelined is
generic (
    SIZE      : positive;
    ITERATIONS : positive;
    RESET_ACTIVE_LEVEL : std_ulogic := '1'
);
port (
    Clock : in std_ulogic;
    Reset : in std_ulogic;

    Mode : in cordic_mode;

    X : in signed(SIZE-1 downto 0);
    Y : in signed(SIZE-1 downto 0);
    Z : in signed(SIZE-1 downto 0);

    X_result : out signed(SIZE-1 downto 0);

```

```

    Y_result : out signed(SIZE-1 downto 0);
    Z_result : out signed(SIZE-1 downto 0)
);
end entity;

architecture rtl of cordic_pipelined is
    type signed_pipeline is array (natural range <>) of signed(SIZE-1 downto 0);

    signal x_pl, y_pl, z_pl : signed_pipeline(1 to ITERATIONS);
    signal x_array, y_array, z_array : signed_pipeline(0 to ITERATIONS);

    function gen_atan_table(size : positive; iterations : positive) return signed_pipeline is
        variable table : signed_pipeline(0 to ITERATIONS-1);
    begin
        for i in table'range loop
            table(i) := to_signed(integer(arctan(2.0**(-i)) * 2.0**size / MATH_2_PI), size);
        end loop;

        return table;
    end function;

    constant ATAN_TABLE : signed_pipeline(0 to ITERATIONS-1) := gen_atan_table(SIZE, ITERATIONS);
begin

    x_array <= X & x_pl;
    y_array <= Y & y_pl;
    z_array <= Z & z_pl;

    cordic: process(Clock, Reset) is
        variable negative : boolean;
    begin
        if Reset = RESET_ACTIVE_LEVEL then
            x_pl <= (others => (others => '0'));
            y_pl <= (others => (others => '0'));
            z_pl <= (others => (others => '0'));

        elsif rising_edge(Clock) then
            for i in 1 to ITERATIONS loop
                if Mode = cordic_rotate then
                    negative := z_array(i-1)(z'high) = '1';
                else
                    negative := y_array(i-1)(y'high) = '0';
                end if;

                --if z_array(i-1)(z'high) = '1' then -- z is negative
                if negative then
                    x_pl(i) <= x_array(i-1) + (y_array(i-1) / 2**(i-1));
                    y_pl(i) <= y_array(i-1) - (x_array(i-1) / 2**(i-1));
                    z_pl(i) <= z_array(i-1) + ATAN_TABLE(i-1);
                else -- z or y is positive
                    x_pl(i) <= x_array(i-1) - (y_array(i-1) / 2**(i-1));
                    y_pl(i) <= y_array(i-1) + (x_array(i-1) / 2**(i-1));
                    z_pl(i) <= z_array(i-1) - ATAN_TABLE(i-1);
                end if;
            end loop;
        end if;
    end process;
end architecture;

```

```

        end if;
    end process;

    X_result <= x_array(x_array'high);
    Y_result <= y_array(y_array'high);
    Z_result <= z_array(z_array'high);

end architecture;

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

library project;
use project.cordic.all;
--use extras.pipelining.all;

entity cordic_flex_pipelined is
    generic (
        SIZE          : positive;
        ITERATIONS     : positive;
        PIPELINE_STAGES : natural;
        RESET_ACTIVE_LEVEL : std_ulogic := '1'
    );
    port (
        Clock : in std_ulogic;
        Reset : in std_ulogic;

        Mode : in cordic_mode;

        X : in signed(SIZE-1 downto 0);
        Y : in signed(SIZE-1 downto 0);
        Z : in signed(SIZE-1 downto 0);

        X_result : out signed(SIZE-1 downto 0);
        Y_result : out signed(SIZE-1 downto 0);
        Z_result : out signed(SIZE-1 downto 0)
    );
end entity;

architecture rtl of cordic_flex_pipelined is
    signal x_result_pre : signed(X'range);
    signal y_result_pre : signed(Y'range);
    signal z_result_pre : signed(Z'range);
begin

    m: process(Mode, X, Y, Z) is
    begin
        if Mode = cordic_rotate then
            rotate(ITERATIONS, X, Y, Z, x_result_pre, y_result_pre, z_result_pre);
        else
            vector(ITERATIONS, X, Y, Z, x_result_pre, y_result_pre, z_result_pre);
        end if;
    end process;
end architecture;

```



```

pl_x: pipeline_s
generic map (
    PIPELINE_STAGES => PIPELINE_STAGES,
    RESET_ACTIVE_LEVEL => RESET_ACTIVE_LEVEL
) port map (
    Clock => Clock,
    Reset => Reset,
    Sig_in => x_result_pre,
    Sig_out => X_result
);

```

```

pl_y: pipeline_s
generic map (
    PIPELINE_STAGES => PIPELINE_STAGES,
    RESET_ACTIVE_LEVEL => RESET_ACTIVE_LEVEL
) port map (
    Clock => Clock,
    Reset => Reset,
    Sig_in => y_result_pre,
    Sig_out => Y_result
);

```

```

pl_z: pipeline_s
generic map (
    PIPELINE_STAGES => PIPELINE_STAGES,
    RESET_ACTIVE_LEVEL => RESET_ACTIVE_LEVEL
) port map (
    Clock => Clock,
    Reset => Reset,
    Sig_in => z_result_pre,
    Sig_out => Z_result
);

```

```

end architecture;

```

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.math_real.all;

```

```

library project;
use project.cordic.all;

```

```

entity cordic_sequential is
generic (
    SIZE      : positive;
    ITERATIONS : positive;
    RESET_ACTIVE_LEVEL : std_ulogic := '1'
);
port (
    Clock : in std_ulogic;
    Reset : in std_ulogic;

```

```

Data_valid : in std_ulogic; --# Load new input data
Busy      : out std_ulogic; --# Generating new result
Result_valid : out std_ulogic; --# Flag when result is valid
Mode      : in cordic_mode; --# Rotation or vector mode selection

X : in signed(SIZE-1 downto 0);
Y : in signed(SIZE-1 downto 0);
Z : in signed(SIZE-1 downto 0);

X_result : out signed(SIZE-1 downto 0);
Y_result : out signed(SIZE-1 downto 0);
Z_result : out signed(SIZE-1 downto 0)
);
end entity;

architecture rtl of cordic_sequential is
type signed_array is array (natural range <>) of signed(SIZE-1 downto 0);

function gen_atan_table(size : positive; iterations : positive) return signed_array is
variable table : signed_array(0 to ITERATIONS-1);
begin
for i in table'range loop
table(i) := to_signed(integer(arctan(2.0**(-i)) * 2.0**size / MATH_2_PI), size);
end loop;

return table;
end function;

constant ATAN_TABLE : signed_array(0 to ITERATIONS-1) := gen_atan_table(SIZE, ITERATIONS);

signal xr : signed(X'range);
signal yr : signed(Y'range);
signal zr : signed(Z'range);

signal x_shift : signed(X'range);
signal y_shift : signed(Y'range);

subtype iter_count is integer range 0 to ITERATIONS;

signal cur_iter : iter_count;
begin

cordic: process(Clock, Reset) is
variable negative : boolean;
begin
if Reset = RESET_ACTIVE_LEVEL then
xr <= (others => '0');
yr <= (others => '0');
zr <= (others => '0');
cur_iter <= 0;
Result_valid <= '0';
Busy <= '0';
elsif rising_edge(Clock) then
if Data_valid = '1' then

```

```

xr <= X;
yr <= Y;
zr <= Z;
cur_iter <= 0;
Result_valid <= '0';
Busy <= '1';
else
if cur_iter /= ITERATIONS then
--if cur_iter(ITERATIONS) /= '1' then
if Mode = cordic_rotate then
negative := zr(z'high) = '1';
else
negative := yr(y'high) = '0';
end if;

--if zr(z'high) = '1' then -- z or y is negative
if negative then
xr <= xr + y_shift; --(yr / 2**(cur_iter));
yr <= yr - x_shift; --(xr / 2**(cur_iter));
zr <= zr + ATAN_TABLE(cur_iter);
else -- z or y is positive
xr <= xr - y_shift; --(yr / 2**(cur_iter));
yr <= yr + x_shift; --(xr / 2**(cur_iter));
zr <= zr - ATAN_TABLE(cur_iter);
end if;

cur_iter <= cur_iter + 1;
--cur_iter <= '0' & cur_iter(0 to ITERATIONS-1);
end if;

if cur_iter = ITERATIONS-1 then
Result_valid <= '1';
Busy <= '0';
end if;
end if;

end if;
end process;

x_shift <= shift_right(xr, cur_iter);
y_shift <= shift_right(yr, cur_iter);

X_result <= xr;
Y_result <= yr;
Z_result <= zr;

end architecture;

```